Studies of networking algorithms in wireless sensor networks

Zhang, Pengfei

2015

Zhang, P. (2015). Studies of networking algorithms in wireless sensor networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/62910>

<https://doi.org/10.32657/10356/62910>

STUDIES OF NETWORKING ALGORITHMS IN WIRELESS SENSOR NETWORKS

ZHANG PENGFEI

SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING

A thesis submitted to the Nanyang Technological University

in partial fulfillment of the requirement for the degree of

Doctor of Philosophy

2015

Abstract

In traditional wireless sensor networks (WSNs), sensors powered by battery are largely deployed in a field to observe certain physical phenomena, such as temperature, noise and many more. The purpose of WSN is to be able to sense from environment and detect events for a long time. By improving the network lifetime, WSNs are able to detect events longer. This agrees with the nature of WSNs, which is sense-to-detect events through long operations (hopefully) of batteries. However, limited battery capacity and low event detection probability have always been imposing serious challenges and constraints to network applications, stimulating the developments of various new technologies. Existing works on maximizing lifetime include networking clustering technologies which let most sensors communicate with the nearby cluster head (CH) rather than the remote base station; and energy-harvesting (EH) sensors which allow battery recharging by collecting energy from surrounding environment, etc. Existing works on maximizing event detection probability study the case of deterministic and known sensors deployment, where sensors transmit their local observations or decisions over perfectly known channels to a Gateway, which collects the observations and fuses them in order to perform event detection.

In this thesis, we propose a centralized approach to finding the optimal locations for cluster heads and energy harvesting nodes, where we consider nodes powered by ambient energy harvesting as dedicated relay nodes for cluster heads, and propose joint clustering and relay node placement algorithms for network lifetime maximization. In addition, we also propose a distributed EH-CH matching algorithm with given locations for CHs and EHs. To achieve good EH-CH matching in a distributed manner, we borrow ideas adopted in virtual-output-queued (VOQ) networks to develop a simple matching scheme. For the problem of event detection, we develop the first reported solution for wireless sensor network under random deployment. We consider the practical case where the spatial distribution of the sensors follows a Finite

Binomial Point Process (FBPP) and Infinite Poisson Point Process (IPPP) for both homogeneous and inhomogeneous cases. In addition, we present novel distributed event detection algorithms based on a statistical approach that tolerates Byzantine attacks and nonlinear transmission schemes where malicious (compromised) or nonlinear sensors send false sensing data to the gateway leading to increased false alarm rate using Moment Matching and Series Expansion methods. We provide analytical and extensive simulation results to demonstrate the optimality and convergence of all these algorithms, and therefore help provide useful benchmarks for various centralized and distributed scheme designs.

Acknowledgments

I would like to express my sincere thanks and appreciation to my supervisor, Prof. Xiao Gaoxi (NTU), and co-supervisor, Dr. Tan Hwee Pink (I²R) for their invaluable guidance, support and suggestions. Their knowledge, suggestions, and discussions have helped me become a capable researcher. Their encouragement also helped me to overcome the difficulties encountered in my research. I also want to thank my colleagues in Sense & Sense-ability lab of I²R, for their generous help. I want to thank Dr. Ido Nevat for his explanation of the statistical modeling, I also want to thank Dr. Gareth Peters from University of College London for the valuable discussions and contributions towards my papers.

I also want to thank Lin Shaowei and Jing Yang for helping me discuss problem and revise papers. My gratitude also goes to Pius Lee, Jin Yunye, Wong Liangze for their friendship and support. I am very grateful to the members of our Infinitus Lab in NTU. It is a pleasure to collaborate with my teammate, Yu Yi.

I am also indebted to my friend Zhang Li, for his technical and personal suggestions, especially for his help on PPT slides.

Last but not least, I want to thank my wife, Ning Haoyan, and my family in China, for their constant love and encouragement.

Contents

Abstract	i
Acknowledgments	iii
List of Figures	viii
List of Tables	xii
List of Abbreviations	xiii
1 Introduction	1
1.1 Background and Motivation	1
1.1.1 Constraints of WSNs	1
1.1.2 Challenging Problems in WSNs	3
1.2 Lifetime Maximization Problem	5
1.3 Event Detection Problem	8
1.4 Outline	12
1.5 Publications	14
2 Literature Review	16
2.1 Lifetime Maximization of WSN	16
2.1.1 Lifetime Definitions	16
2.1.2 Clustering Algorithms in Battery Powered WSN	17
2.1.3 Clustering Algorithms in EHWSN	23
2.1.4 Non-Clustering Algorithms in WSN	25
2.1.5 Relay Node Placement in WSNs	27
2.2 Event detection in Wireless Sensor Networks	28
2.2.1 Distributed Event Detection Algorithms in Deterministic WSNs	29
2.2.2 Distributed Event Detection Algorithms in Random WSNs	31

2.2.3	Distributed Event Detection Algorithms in WSNs with Malicious Users	32
2.3	Statistical Decision Theory	34
2.4	Series Expansions Methods	36
3	Centralized Lifetime Maximization Clustering Algorithms for WSNs	40
3.1	System Model	40
3.2	Lifetime Maximization Clustering Algorithms	41
3.2.1	Cluster Head Selection Algorithm ($N_c = 1, \alpha = \frac{1}{N_s}$)	42
3.2.2	Cluster Formation Algorithm ($N_c > 1, \alpha = \frac{1}{N_s}$)	45
3.2.3	Dynamic Clustering Algorithm ($N_c \geq 1, \alpha = \frac{1}{N_s}$)	48
3.3	Algorithms for Maximizing Lifetime of WSNs with EH Sensors ($\alpha = \frac{1}{N_s}$)	50
3.4	Algorithms for Maximizing Lifetime of WSNs with EH Sensors ($\frac{1}{N_s} < \alpha < 1$)	54
3.5	Simulation Results	56
3.5.1	Performance of Clustering Algorithms without EH Nodes	57
3.5.2	Performance of Joint EH placement and Clustering Algorithm	60
3.5.3	Algorithm Performance for Different Lifetime Definitions	66
3.6	Conclusion	66
4	Distributed Clustering Algorithms for Maximizing Lifetime	68
4.1	Lifetime-Optimal Single Battery Powered Relay Node Scheduling for Multi-clustered WSN	68
4.1.1	System Model	68
4.1.2	Distributed Iterative Algorithm for Relay Node Scheduling	70
4.2	Distributed Clustering Algorithms for Multiple-clustered WSN with Multiple Energy Harvesting Relay Nodes	74
4.2.1	Network Model	74
4.2.2	Distributed Clustering Algorithm for Maximizing Lifetime in WSNs with EH Sensors	77
4.2.3	Benchmark Matching Algorithms for Maximizing Lifetime of WSNs with EH Sensors	82
4.3	Simulation Results	85

4.3.1	Performance of Distributed Clustering Algorithms with Single Battery Powered Relay	85
4.3.2	Performance of Distributed Clustering Algorithms with Multiple Energy Harvesting Relay Nodes	88
4.4	Conclusion	94
5	Event Detection Algorithms in Random WSNs	95
5.1	System Model and Problem Formulation	95
5.1.1	Wireless Sensor Network Operation Model	98
5.1.2	Optimal Event Detection Decision Rule	99
5.2	Event Detection Algorithm under Random Point Process Sensor Deployment .	100
5.3	Probability Density Approximation via Series Expansion Methods	102
5.3.1	Gram-Charlier Series Expansion	103
5.3.2	Gamma-Laguerre Series Expansion	104
5.3.3	Beta-Jacobi Series Expansion	106
5.4	Calculation of The Moments	108
5.5	Simulation Results	109
5.5.1	Performance of Event Detection Algorithm using Gram-Chalier Series Expansion	110
5.5.2	Moments Calculation	113
5.5.3	Comparison of Critical Region	116
5.5.4	Marginal Likelihood Estimation via Series Expansion and S-K Region .	116
5.5.5	Event Detection Performance Comparison	120
5.6	Conclusions	121
6	Event Detection Algorithms with Incomplete Information from Sensors	124
6.1	Event Detection Algorithms in WSNs with Byzantine Attacks	124
6.1.1	Sensor Network Model	124
6.1.2	Optimal Detection Decision Rule	127
6.1.3	Unknown CSI Between Source to Sensors and Perfect CSI Between Sensors to GW	129

6.1.4	Unknown CSI Between Source to Sensors and Unknown CSI Between Sensors to GW	130
6.2	Event Detection Algorithms in WSNs with Nonlinear Transmission	132
6.2.1	Wireless Sensor Network System Model and Definitions	132
6.2.2	Event Detection via Beta-Jacobi Density Series Expansion	135
6.2.3	Calculation of The Moments	135
6.3	Simulation Results	138
6.3.1	Event Detection Algorithm with Byzantine Attack	138
6.3.2	Distributed Event Detection Algorithm Nonlinear Transmission	142
6.4	Conclusion	144
7	Summary and Future Work	146
7.1	Summary	146
7.2	Future Work	147
A	Appendix	150
A.1	Proof of Theorem 5.1	150
A.2	Proof of Lemma 5.1	151
A.3	Proof of Theorem 5.2	152
A.4	Proof of Theorem 5.3	152
A.5	Proof of Theorem 5.4	154
A.6	Proof of Lemma 5.5	155
A.7	Proof of Theorem 6.1	157
	References	159

List of Figures

1.1	A two tier hierarchical architecture for WSN.	5
1.2	Event Detection System Model: (a) Source to sensors (b) Sensors to GW . . .	8
2.1	Abstract model of optical VOQ switch [1].	25
2.2	Density of true PDF and approximated PDF using Edgeworth Series Expansions.	39
3.1	The single-cluster WSN model.	42
3.2	Our proposed multiple cluster WSN model.	46
3.3	Step 1-2 of cluster formation algorithm	47
3.4	Step 3-4 of cluster formation algorithm	47
3.5	Cluster head selection + cluster formation algorithm	48
3.6	Comparison of convergence efficiency of our algorithm for all cases ($N_s = 100$).	58
3.7	Comparison between our proposed algorithm with selected algorithms for (a) Case I; (b) Case II.	60
3.8	Network lifetime vs number of clusters for our proposed algorithm ($N_c =$ $\{1,2,\dots,10\}$).	61
3.9	Effects of using EH node on the network lifetime ($N_c = 1, P_{EH,h} = 0.03W$). . .	62
3.10	Network lifetime vs number of clusters for our proposed algorithm ($N_c =$ $\{1,2,\dots,10\}, P_{EH,h} = 0.03W$).	63
3.11	Effects of energy harvesting rate on network lifetime for different cases with optimal number of clusters for (a) Case I; (b) Case II.	64
3.12	Effects of energy harvesting rate on network lifetime using dynamic clustering algorithm for (a) Case I; (b) Case II.	65
3.13	Comparison of our algorithm with brute force method at various α (Case II, N_c $= 3, N_s = \{100,150\}$)	66

4.1	Our proposed distributed clustered WSN model.	70
4.2	Abstract model for lifetime expression.	71
4.3	Abstract model of multiple EH nodes serve as relay for CHs.	75
4.4	Typical energy profile of EH node with time.	76
4.5	Overall flow chart for our design.	76
4.6	The average numbers of iterations required to converge while using Algorithm 4.2 in WSNs with different numbers of EHs/CHs.	81
4.7	Timing diagram for message exchanges in overhead.	81
4.8	Number of survival rounds of each CH for (a) Case I and Area I (b) Case II and Area I	87
4.9	(a) Comparison of network lifetime for six different cases (b) Number of rounds relay node serves for each CH	87
4.10	Effects of number of clusters on network lifetime in Case I ($N_e = 4, P_{eh,h} =$ $0.04W, t_r = T, E_{th} = E_s$).	89
4.11	Effects of t_r on network lifetime in Case I ($p_{s_i} = 0.04, N_e = 4, P_{eh,h} = 0.04W,$ $E_{th} = E_s$)	90
4.12	Effects of number of EH nodes on the network lifetime in Case I ($p_{s_i} = 0.04,$ $P_{eh,h} = 0.04W, t_r = T, E_{th} = E_s$).	90
4.13	Effects of EH node harvesting rate on the network lifetime in Case I ($N_e = 4,$ $p_{s_i} = 0.04, t_r = T, E_{th} = E_s$).	91
4.14	Effects of E_{th} on the network lifetime in Case I ($N_e = 4, p_{s_i} = 0.04, t_r = T,$ $P_{eh,h} = 0.04W$).	92
4.15	Comparison between our algorithm with benchmark algorithms in Case I ($p_{s_i} =$ $0.04, N_e = 4, t_r = T, P_{eh,h} = 0.04W, E_{th} = E_s$).	93
5.1	Realisations from Homogenous BPP deployment (top left), Non-homogenous BPP deployment (bottom left), Homogenous PPP deployment (top right) and Non-homogenous PPP deployment (bottom right)	98
5.2	Probability density estimation of $p(Y(s) \mathbf{X}_s, \mathcal{H}_1)$ using Monte Carlo simula- tion and Gram-Chalier series expansion	111
5.3	ROC performance under different configurations of σ_w	112

5.4	ROC performance under different configurations of P_0	112
5.5	ROC performance under different configurations of radius R	113
5.6	ROC performance under different configurations of propagation loss coefficient α	113
5.7	Comparison of the theoretical results of the first four moments per Theorem 5.4 and the corresponding Monte Carlo simulation results under BPP type deployment: (a) Homogenous BPP deployments. The parameters used: $R = 100, \alpha = 0.5, N = 10, \nu = \{0, 0.5\}$; (b) Non-homogeneous BPP deployments. The parameters used: $R = 100, \alpha = 0.5, \nu = \{0, 0.5\}$	114
5.8	Comparison of the theoretical results of the first four moments per Theorem 5.4 and the corresponding Monte Carlo simulations under PPP type deployments: (a) Homogeneous PPP deployments. The parameters used: $R = 100, \alpha = 0.5, N = 10, \nu = \{0, 0.5\}$; (b) Non-homogeneous PPP deployments. The parameters used: $R = 100, \alpha = 0.5, \nu = \{0, 0.5\}$	115
5.9	Effect of critical region on marginal likelihood estimation as a function of the critical region: (a) Four different sensor deployments with $r = \{0.1, 0.5, 1, 5\}, R = 100, \alpha = 2, \nu = 0, \sigma_w = 0, P_0 = 1, \theta = 2, \eta = 2$; (b) PDF estimation via Gram-Charlier, Gamma-Laguerre and Beta-Jacobi and Monte Carlo simulation.	117
5.10	Gram-Charlier series expansion for three different system parameters presented in Table 5.1: (a) Skew-Kurtosis curves; (b) PDF estimation vs. Monte Carlo simulation.	118
5.11	Gamma-Laguerre series expansion for three different system parameters in Table 5.1: (a) Skew-Kurtosis curves; (b) PDF estimation vs. Monte Carlo simulation.	119
5.12	Beta-Jacobi series expansion for three different system parameters in Table 5.1: (a) Skew-Kurtosis curves; (b) PDF estimation vs. Monte Carlo simulation.	120
5.13	ROC curves for the three series expansions for different values of path-loss exponent $\alpha = \{2, 2.2, 2.4\}$	122
5.14	ROC curves for the three series expansions for different values of path-loss exponent $\alpha = \{1.9, 2.1, 2.3\}$	123

6.1	Sensor network model with M sensors over wireless channels. Each sensor may be compromised by a malicious (Byzantine) attack $\mathcal{T}_B(R_m(l))$	125
6.2	Wireless sensor network with M sensors over wireless channels.	132
6.3	Case I - ROC performance given unknown CSI between the source and sensors and perfect CSI between sensors and the GW under different configurations of the number of malicious sensors $\Pr(\mathcal{H}_B^s) \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.55, 0.6\}$, $(M = 100, L = 1)$	139
6.4	Case I - ROC performance given unknown CSI between the source and sensors and perfect CSI between sensors and the GW under different configurations of $\bar{\mathbf{F}} \in \{0.1 + 0.1i, \dots, 0.5 + 0.5i\}$, $(\Pr(\mathcal{H}_B^s) = 0.3, M = 100, L = 1)$	139
6.5	Case II - ROC performance under unknown CSI between the source and sensors and unknown CSI between sensors and the GW under different configurations of the number of malicious sensors $\Pr(\mathcal{H}_B^s) \in \{0, 0.1, \dots, 0.6\}$, $(M = 100, L = 1)$	140
6.6	Case II - ROC performance under unknown CSI between the source and sensors and unknown CSI between sensors and the GW under different configurations of $\bar{\mathbf{F}} \in \{0.1 + 0.1i, \dots, 0.45 + 0.45i\}$, $(M = 100, \Pr(\mathcal{H}_B^s) = 0.3, L = 1)$	141
6.7	Case I - ROC performance under unknown CSI between the source and sensors and known CSI between sensors and the GW for different schemes $\bar{\mathbf{F}} = 0.2 + 0.2i$, $(M = 100, \Pr(\mathcal{H}_B^s) = 0.3, L = 1)$	141
6.8	ROC curve with respect to different number of sensors $M \in \{25, 35, 45\}$. Other parameters: $p_l = 0.5$ $P_{max} = 0.1$, $\sigma_w = \sigma_v = 0.1$, $\sigma_H = \sigma_G = 0.1$, $\mu_H = \mu_G = 0.2$ $K = 1$, $\theta = 1$	142
6.9	ROC curve with respect to different proportion of linear vs. nonlinear sensors $p_l \in \{1, 0.8, 0.4\}$. $M = 30$, $P_{max} = 0.1$, $\sigma_w = \sigma_v = 0.1$, $\sigma_H = \sigma_G = 0.1$, $\mu_H = \mu_G = 0.2$ $K = 1$, $\theta = 1$	143
6.10	ROC curve with respect to different $P_{max} \in \{1, 0.8, 0.6\}$. $p_l = 0.5$. $M = 30$, $\sigma_w = \sigma_v = 0.1$, $\sigma_H = \sigma_G = 0.1$, $\mu_H = \mu_G = 0.2$	144

List of Tables

2.1	Categories of Algorithms	17
2.2	Categorization of network lifetime definitions [2]	17
2.3	Network lifetime compared to standard LEACH [3]	19
2.4	Comparison of energy harvesting rates of various energy harvesting techniques [4]	26
3.1	Notations used throughout this chapter	41
3.2	Comparison of network lifetime for various cases ($N_s = 100, N_c = 1$)	57
3.3	Comparison between our method and brute force method ($N_c = \{2,3\}, N_s =$ $\{100,150\}$)	58
4.1	Notations used throughout section 4.1	69
4.2	Notations used throughout section 4.2	77
5.1	Series Expansion Parameters	118

List of Abbreviation

WSN	Wireless Sensor Networks
BS	Base Station
CH	Cluster Head
NCH	non Cluster Head
EHWSN	Energy Harvesting WSN
LP	Linear Programming
CSI	Channel State Information
GW	Gateway
LEACH	Low-Energy Adaptive Clustering Hierarchy
MEMS	Micro-Electro-Mechanical Systems
IoT	Internet of Things
MAC	Medium Access Control
VOQ	Virtual Output Queuing
SNR	Signal-to-Noise-Ratio
PPP	Poisson Point Process
BPP	Binomial Point Process
LRT	Likelihood Ratio test
S-K	Skew-Kurtosis
EECS	Energy Efficient Clustering Scheme
HEED	Hybrid Energy Efficient Approach
LEACH-C	LEACH-Centralized
MTE	Minimum-Transmission-Energy
SEP	Stable Election Protocol
LEACH-E	Enhanced-LEACH
MINLP	Mixed Integer Nonlinear Programming
MDEA	Micro-Server Deployment and Energy Allocation
MIS	Maximum Independent Set
DIS	Dominating Independent Set
QoS	Quality of Service
SCH	Super-Cluster-Head
UFLP	Uncapacitated Facility Location Problem
LDS	Linear-Distance Scheduling

E-EEHF	Enhanced-Environmental Energy Harvesting Framework
PC-sLEACH	Partition-based Centralized sLEACH
CSMA	Carrier Sense Multiple Access
PIM	Parallel Iterative Matching
SOS	Self Organizing Sensor
ACE	Algorithm for Cluster Establishment
CMA	Cluster Merging Algorithm

Chapter 1

Introduction

1.1 Background and Motivation

A wireless sensor network (WSN) consists of hundreds or even thousands of autonomous, low-powered and multi-functional sensors to sense data (such as temperature, sound, vibration level and pressure etc) from the environment. A sensor typically consists of wireless communication modules, control processing units (CPU), and energy storage devices such as batteries or super-capacitors. With the development of advanced wireless communication technologies and Micro-Electro-Mechanical Systems (MEMS) technologies, low-powered/small-sized and multi-functional sensors have replaced the traditional sensors. WSNs have been widely used in many practical applications [5–8], including health monitoring, lifestyle enrichment, smart vehicles monitoring, building maintenance, structural health monitoring etc. WSNs will revolutionize the way people work or live in the near future. WSNs are expected to be accessible through the Internet, revolutionizing the way people work or live, in the near future.

However, there are numerous constraints on sensors such as limited energy availability, small bandwidth, low computational capabilities, data misinterpretation, and the vulnerability to malicious attacks, requiring thorough and prudent research effort to overcome these challenges.

1.1.1 Constraints of WSNs

Although a large number of protocols have been devised and applied to wireless networks, those protocols/algorithms cannot be directly applied to WSNs since WSNs possess special

characteristics and constraints which distinguish them from other wireless networks. These special characteristics and constraints include:

- (i) *Limited Energy*: Due to the small sizes and irreplaceable properties of sensor batteries (i.e., when sensors are deployed in a remote area), sensors may have limited energy supply. Therefore, energy conservation is always among the most critical requirements of WSN protocol design.
- (ii) *Limited Computational Capability*: WSNs are usually composed of sensors with low computational capabilities. But the tasks required for WSNs are quite challenging and may require dynamic collaborations between nodes. This limits the algorithm types that can be used in sensors.
- (iii) *Dynamic Topology*: WSNs are deployed in many diverse types of environments, that may be harsh to operate in, such as underwater or in volcanoes. In those cases sensors may become faulty, may deplete their energy resources or may be compromised. These will have a profound impact on the connectivity between sensors which can be time varying.
- (iv) *Location Uncertainty*: Since sensors are deployed in a complex natural environment which may not be accessible by humans, it can be very hard to precisely know the locations of sensors. In many cases the sensors are not equipped with a GPS device or if installed in an environment without a clear Line-of-Sight, the obstacles may deteriorate the quality of the localization performance. This can have a significant negative impact on many applications such as field reconstruction where the locations of the sensors are required to have a high quality estimation of the field.
- (v) *Fluctuating Wireless Channels*: In WSNs, sensors transmit information through wireless channels. Wireless channels are random and may fluctuate very fast, leading to transmission loss in the form of packet loss. Designing robust schemes to handle the decremental effects of the wireless channel is therefore paramount.
- (vi) *Inaccurate Environment Readings*: Sensors are deployed to observe certain physical phenomenon, such as temperature, noise, humidity, etc. The sensors need to accurately detect certain events when they occur with minimal delay. Examples include detection of a

fire in a forest or there is extreme loud noise in a region. Designing appropriate schemes to handle event detection is one of the most important aspects of sensor networks.

- (vii) *Heterogeneity and Scalability*: WSNs may consist of different types of sensors in terms of computational capabilities and battery sizes. A clustered WSN is composed of different types of sensors, Cluster Heads (CHs)/ Gateway (GW) are more powerful sensors, non-cluster heads (NCHs) are less powerful sensors. Some sensors may have energy harvesting capabilities, while the rest may be battery powered. All these properties contribute to the heterogeneity of WSN. Meanwhile, a large number of heterogeneous sensors may be deployed, which makes it necessary to design schemes with high scalability.

1.1.2 Challenging Problems in WSNs

Due to the various constraints mentioned in Section 1.1.1, many challenging problems exist in WSNs. In this section, we itemize the issues that draw most attention by researchers.

- (i) *Routing*: When a large scale WSN is deployed, it is a big challenge to design adaptive routing schemes to transmit information from sensors to GW. This is due to poor channels, obstacles, limited energy or transmission power, faulty sensors and malicious users.
- (ii) *Algorithms Type*: Due to limited energy/computational capabilities of sensors, simple efficient and fault tolerant algorithms need to be implemented in WSNs. Therefore, each sensor makes local decisions based on distributed algorithms without consuming much energy and computation. On the other hand, centralized algorithms where GW makes global decisions based on information of all sensors, are also useful to provide benchmark for distributed algorithms. Through centralized algorithms, the designers can know the number of sensors to deploy, region of area to deploy, scheduling schemes of sensors, duty-cycle schemes before deployment. It is a challenging problem to design suitable algorithms for various scenarios.
- (iii) *Topology Control*: Due to the large number of sensors, it is a challenging problem to provide robustness to networks with dynamic topologies. It is an even more challenging problem when connectivity/coverage constraints are taken into consideration due to the dynamic topologies.

- (iv) *Localization*: Due to the indoor environments/obstacles where the GPS readings are rendered ineffective, it is a challenging problem to design intelligent localization algorithms.
- (v) *Event Detection*: It is a challenging problem to design appropriate event detection algorithms for fluctuating wireless channels which are common in practical deployments. It is difficult to propose event detection algorithm if the channel conditions are bad.
- (vi) *Field Construction*: Field construction is a challenging problem in WSNs because the information transmitted from all the sensors are noisy. The information from all the sensors have either spatial or temporal correlation. It is a big challenge to estimate the whole field using such properties.
- (vii) *Security*: WSNs have many military or commercial applications. It is important to protect the privacy and security of users. However, sensors are vulnerable to attacks, either by outsiders or internally. One of such attacks is Byzantine attacks [9, 10]. In Byzantine attack, some of the sensors collude to send false information to GW with the purpose of confusing or "blinding" the GW, leading to incorrect decisions. It is a challenging problem to design secure schemes to protect WSN from such attacks.

We select two critical aspects of these problems to study in this thesis. The first problem we study is the lifetime maximization problem (as described in (i) in section 1.1.1). The second problem we study is the event detection problem (as described in (v) above). The problems are detailed below:

- (i) Limited energy is the most critical problem in wireless sensor networks. With small sizes and limited battery capacity, it may be inconvenient to change battery even when they are not deployed in an inaccessible area. Therefore, we study the lifetime maximization problem, where we use suitable techniques to maximize the WSN lifetime. Specifically, we use clustering approach combined with energy harvesting capabilities.
- (ii) Event detection problem is another fundamental research problem in WSNs. In this thesis, we consider the event detection problems in WSNs with unknown sensors/sources location, non-stationary wireless channels and malicious attacks.

1.2 Lifetime Maximization Problem

Since batteries in sensors have finite stored energy and it is typically not convenient to replace or re-charge these batteries, an important aspect in WSN schemes is to achieve high energy efficiency in order to prolong the lifetime of the network. Research on energy-efficient WSNs has covered broad topics on networking layer, medium access control layer (MAC), and systems design/testing, etc.

Figure 1.1: A two tier hierarchical architecture for WSN.

On the networking layer, an extensively studied topic is to utilize clustering technology (Figure 1.1 shows a simple two tiered hierarchical architecture for WSN) to lower network energy consumption [8, 11]. Specifically, in a clustered network, sensors are grouped into clusters. A clustered WSN comprises of four types of entities: (i) a base station (BS) which is the data sink, and is usually assumed to have infinite energy (i.e., mains powered); (ii) non cluster head (NCH) sensor nodes; (iii) cluster head (CH) sensor nodes; and (iv) relay nodes. The NCHs sense the environment, and forward the data to their respective CHs. CHs sense the environment, receive and aggregate data from NCH nodes, and forward the data to the BS, either directly or through relay nodes. Relay nodes are responsible for forwarding data received from other nodes and may not necessarily be responsible for local sensing. In clustered WSNs, transmitting to a nearby CH rather than to a possibly far away BS helps reduce the energy consumption of NCHs. There are three key phases involved in maximizing the lifetime of such a network: (i) Cluster head selection; (ii) Cluster formation and transmission scheduling;

and (iii) relay node placement and scheduling. By letting each NCH transmit to the nearby CH rather than the possibly far away base station (BS), the energy consumption of the network may be lowered. CHs consume more energy than NCHs due to the additional energy consumed for receiving and aggregating data. Moreover, CHs usually transmit over a longer distance to the relay nodes compared to the distances between NCHs and CHs. Therefore, proper selection of CHs is of utmost importance for prolonging the network lifetime, as a poor choice of CHs may force either NCHs or CHs to use up their energy quickly. An inappropriate selection of CHs may shorten the lifespan of CHs, especially in the absence of relay nodes between CHs and BS. Therefore, lowering the energy consumption of CHs usually plays a critical role in prolonging the lifetime of clustered WSNs. After the CHs are selected, the NCHs select the appropriate clusters to join in order to maximize their lifetime. However, NCHs do not always join the closest CHs since a large cluster size may force CHs to die quickly. Therefore, cluster formation has to be considered jointly with cluster head selection.

Clustered WSNs have been extensively studied in recent years [12]. Existing works include energy-efficient schemes and algorithms [13–18], enhancement of cluster stability in various network topologies [19, 20], MAC layer design [15, 21, 22] and many more. The work on energy-efficient schemes typically adopts two different objectives, namely minimizing the overall energy consumptions [13, 14] and maximizing the network lifetime [23, 24] respectively. These two parts of work are closely related to but different from each other: the former focuses on solving a minimization problem while the latter usually solves a min-max problem since the lifetime of a network is usually decided, or at least strongly affected, by those nodes with shortest lifespan. We term such nodes as the bottleneck nodes. In clustered WSNs, as mentioned earlier, the bottleneck nodes are usually, though not always, CHs. In this thesis, the clustering algorithms are designed to maximize network lifetime.

Existing approaches on lifetime maximization problem can be largely classified into two categories: centralized methods [23–36] and distributed methods [13–15, 37–39]. Centralized methods typically require knowledge of the sensors' locations to achieve global optimization with respect to certain performance metrics. Distributed methods, on the other hand, make decisions based on local information exchanged between neighboring sensors, thus achieving better scalability. Centralized methods are studied in Chapter 3 as they can provide a good

reference for network pre-planning and serve as a useful benchmark for evaluating the performance of distributed methods. Then distributed approaches that make decisions based on local information exchanged between nearby sensors are proposed in Chapter 4.

Parallel to the network clustering techniques, another important recent progress is the development of energy harvesting (EH) sensors [4, 40–43]. EH sensors can harvest energy (e.g., solar, kinetic, thermal etc) from their environment, converting this energy into electrical energy which is then stored in devices with large numbers of recharge cycles (such as super-capacitors) [4] to achieve virtually infinite lifetime. With such developments, it becomes necessary to design new schemes and protocols for EH-WSNs. While the deployment of large-scale WSNs composed solely of EH sensors remains impractical in the near future due to high costs and low achievable duty cycles, an arguably more practical approach is to deploy EH sensor nodes sparsely in WSNs [44, 45].

Clustering methods and energy-harvesting techniques come as a natural combination for prolonging the network lifetime: a proper formation of clusters liberates most sensors (especially NCHs) from high energy consumption, while a carefully planned sparse deployment of EH sensors helps prolong the lifespan of the bottleneck nodes. Since energy harvesting rates are sensitive to the environment [43], it may not be practical to let EH nodes serve as function-critical nodes such as the CHs. In this thesis, we consider a simple case where EH sensors serve as relay nodes for the CHs. By communicating with EH nodes over a shorter distance rather than sending data to BS directly, CHs lowered their energy consumptions for at least a certain fraction of time (e.g., when the EH nodes are up and working). This simple case provides some useful insights into where EH sensors should be located in order to maximize network lifetime.

1.3 Event Detection Problem

1.2.a

1.2.b

Figure 1.2: Event Detection System Model: (a) Source to sensors (b) Sensors to GW

Next, some fundamental studies on event detection algorithms are performed in WSNs. WSNs consist of a set of spatially distributed sensors which monitor a spatial physical phenomenon containing some desired attributes (e.g., pressure, temperature, concentrations of substance, sound intensity, radiation levels, pollution concentrations etc.) as shown in Fig. 1.2.a, and regularly communicate their observations to a Gateway (GW) [46–48] as shown in Fig. 1.2.b. The GW collects these observations and fuses them in order to perform event detection, based on which effective actions can be made [8]. The detection problem of the WSN can be cast as distinguishing between two hypotheses, namely the absence (Null Hypothesis), or presence (Alternative Hypothesis) of a certain event [49–51]. The ability of a WSN to perform such

detection and decisions is crucial for various applications, for example the detection of the presence or absence of a target in a surveillance system, detection of missiles, detection of chemical, biological or nuclear plumes and many more [52–54]. It is therefore imperative for the WSN to be accurate in detecting a valid event (high detection rate) while maintaining as low as possible false detection (low false alarm).

Previous works on event detection have concentrated on cases where the sensors deployment (i.e., the locations of the sensors) is deterministic and known to the GW ([8, 55–58] and references within). For example, in [59], the problem of distributed detection was considered, where the sensors transmit their local decisions over perfectly known wireless channels. In [60], the problem of distributed event detection under Byzantine attack was considered. Theoretical performance analysis was derived in [57] for detection fusion under conditionally dependent and independent local decisions. Distributed detection in WSNs over fading channels with multiple receive antennas at the Gateway (GW) was considered in [58]. In contrast, the problem of event detection where the sensors are randomly deployed in the field has not been addressed before. This problem is of great practical interest because in many cases the locations of the sensors are unknown to the GW. The following are examples of such scenarios:

- (i) Surveillance: sensor nodes are dropped by airplanes, unmanned aerial vehicles or ships in order to survey a region of interest that is inaccessible from the ground. To increase life span and to reduce costs of the sensors, they are not equipped with localization device (i.e., GPS) and their location is considered random and unknown [61, 62].
- (ii) Privacy-preserving participatory sensing: individuals share certain environmental information (e.g., temperature readings, traffic conditions,, etc.) to produce aggregated models. In order to protect their privacy, the users do not share their location information [63].

In addition, in the wireless communication literature there has been great interest in random deployments of wireless networks, see for example [64, 65]. These works make use of tools from stochastic geometry to calculate parameters of interest, such as capacity, Signal-to-Noise-Ratio (SNR) of such systems. These works mainly consider homogeneous deployments, mainly due to the mathematical tractability. In practice however, it's very unlikely that the sensors would

be distributed in space in a spatially homogenous way, but instead a non-homogeneous behavior is more likely to occur, see [66, 67].

To address these two practical aspects of random deployment of inhomogeneous spatially deployed sensor networks, new models and algorithms for event detection need to be developed. In addition, it is important for network designers to understand how different parameters would affect the performance of the WSN before they deploy the WSN, (i.e., number of sensors, region of deployment, level of inhomogeneity of the deployment, etc.) in order to obtain optimal detection performance. It is therefore important to study the average performance of a network before a decision is made.

At the heart of distance based algorithms in WSN under random spatial deployment lies the understanding of distance distribution. Such quantities have been derived under spatial deployment such as Poisson Point Process (PPP) and Binomial Point Process (BPP) [68–71]. In [68], sensors are uniformly randomly distributed following a Binomial Point Process. The authors analyzed various properties of such networks including the distance distribution, moments of distance etc. In [69], the authors consider a more general distribution, namely the Poisson Point Process (PPP) and provide analysis of the distance distributions for such networks. In [70], the authors discuss the deployment of cognitive cellular wireless networks. In [71], the authors considered distance distributions on mobile wireless networks. It is important to note that these papers have only tackled homogeneous type deployments and the practical cases of non-homogeneous deployments have not been addressed.

In this thesis, a novel event detection algorithm is developed in WSN for the case where the sensors are randomly distributed in space and their locations are unknown to the gateway (GW). When the target (event) is present/active, it emits energy (acoustic or electromagnetic) which is measured by each of the sensors. All the measurements from the sensors are then aggregated to the GW which makes the final decision whether the target is present or absent. We assume an energy decay model in which the amount of energy each sensor measures falls off with distance obeying an inverse power-law where the exponent is known as the path loss exponent [65]. In contrast to previous works which assumed that the locations of the sensors are known to the GW [49, 58, 72], we assume a random spatial deployment, meaning that the distance from the target to the sensors is now a random variable.

To obtain the optimal decision rules, the likelihood ratio test (LRT) for the two hypotheses (event present/ absent) needs to be evaluated. This involves the calculation of the marginal

likelihood density under each of the hypotheses. While deriving the marginal likelihood under the Null hypothesis is trivial, the derivation of the marginal likelihood under the Alternative hypothesis is not readily obtained in closed-form, since it involves a multi-variate convolution which cannot be solved exactly. As such, a principled approach is adopted instead to approximate the marginal likelihood under the alternative hypothesis. To do so, stochastic geometry techniques are exploited to model the placement of the sensors. Then the intractable distribution density is approximated via series expansions techniques which are based on an Askey-orthogonal polynomial expansion. The three series expansions we develop are the Gram-Charlier, Gamma-Laguerre and Beta-Jacobi type series expansions. Since these expansions do not ensure positivity of the density at all points, it is important to characterise the system parameters for which the density approximation will remain positive. This characterisation can be carried out by finding the appropriate regions in the Skew-Kurtosis plane (S-K plane) which generate positive support [73]. This characterisation is important and should be used as a guide to choosing the appropriate series expansion. We will illustrate the implications of not choosing the correct series expansion via simulations. Importantly, the algorithms that are developed only require deriving the first four cumulants and moments to obtain good detection performance and are therefore of low computational complexity.

In addition to the event detection problem in random networks, the other common purpose of the WSN is to raise alarms when events of interest such as fires or loud noises occur. Network managers will then respond accordingly to the event by sending a specialist to physically verify and rectify the event. Consequently, it is important to correctly identify a valid event because it would be inefficient to regularly send a person down to the field due to false events. While there has been much research on the detection of malicious users [74], the detection of events in the presence of malicious users is still largely unexplored. WSNs function in a distributed manner and a large number of sensor nodes may be scattered throughout a wide area. Therefore it will not be cost effective to physically monitor and protect all the sensor nodes. Malfunctioned or compromised nodes may report anomalous data and the gateway must be able to detect such abnormalities. In addition, it needs to determine with high probability if an actual event had occurred. One of the malicious attacks that sensor nodes are vulnerable to is the Byzantine attack [75].

In existing work on event detection for wireless sensor networks, most of them are based on the simplifying assumptions that there is perfect Channel State Information (CSI) between the

sensors and the gateway (GW). In addition, they have not considered the effect of malicious users on the system performance. We include the consideration of malicious user and non perfect CSI in our event detection problem.

In addition to the malicious users, another important aspect of power transmission of the sensors is that the transmitter is linear with constant gain. This assumptions does not hold with conventional amplifiers, which are only linear when operating in the inefficient class-A region of operation and consume a large amount of battery power [76, 77].

In this thesis, we study the effect of nonlinear transmissions functions by the sensors over wireless channels. The sensors map their observations using a probabilistic choice of AF functions, the either a linear or bounded (saturated) function before transmission.

1.4 Outline

In this thesis, we consider both the lifetime maximization and event detection problems in WSNs. Specifically, we list the outline of the thesis below and relate the corresponding chapters to the publications in Section 1.5:

- In Chapter 2, we provide background information and literature survey of related work. We shall cover both clustering algorithms and event detection algorithms. We also cover some background knowledge in statistical decision theory and series expansion methods.
- In Chapter 3, we propose algorithms for maximizing the lifetime of clustered WSNs, with or without EH nodes. Specifically, we assume that a given number of sensors are distributed in a certain area with arbitrary distribution. These sensors can be formed into a given number of clusters under centralized control. Each cluster contains a single CH and a certain number of NCHs. NCHs forward data to their CH. Each CH is responsible for aggregating data from NCHs and forwarding the information to BS, either directly or via a dedicated EH relay node. The EH nodes only serve as relay nodes; they do not collect/process environmental information themselves. Furthermore, each sensor is assumed to be equipped with same amount of energy at the beginning and BS has unlimited energy, e.g., it is mains powered. The main contributions of Chapter 3 are two-fold: (i) we propose efficient algorithms for maximizing the network lifetime of both

single-clustered and multi-clustered WSNs. Analytical and extensive simulation results demonstrate the fast convergence of our proposed algorithms to optimal or suboptimal solutions; (ii) based on the assumption that the locations of the EH nodes can be adjusted in order to maximize network lifetime, we extend the proposed algorithms to handle the case where EH sensors serve as relay nodes for CHs. Extensive simulation results quantify how much EH nodes may help prolong the network lifetime. Finally, we also briefly discuss on the revision of the proposed algorithms under different definitions of network lifetime.

- In Chapter 4, Section 4.1, we first devise a simple yet efficient distributed algorithm to optimally allocate a single relay node to serve the CHs to maximize network lifetime. We assume (i) a single battery-powered relay node serves multiple CHs instead of dedicated relay nodes powered by ambient energy harvesting; and (ii) the location of each node is not known to the BS. In Section 4.2, EH sensors are deployed sparsely and matched with CHs to help them relay data to BS without sensing the environment. This EH-CH matching, which is one of the main focuses of Section 4.2, must be done optimally so that the energy consumption of CHs is reduced by communicating through a shorter distance for a certain fraction of time (e.g., when EH nodes are up and working) to relay instead of direct communication with BS. Specifically, we propose a distributed EH-CH matching algorithm with given locations for CHs and EHs. To achieve good EH-CH matching in a distributed manner, we borrow the idea adopted in virtual-output-queued (VOQ) networks [78, 79] to develop a simple matching scheme, and provide analytical and extensive simulation results to demonstrate its fast convergence. We also benchmark its performance against algorithms based on LP and bipartite graph techniques, as well as randomized approach. Simulation results demonstrate much better performance (around 63.55% longer lifetime) compared with randomized approach.
- In Chapter 5, we propose efficient event detection algorithms to deal with two popular deployment scenarios, both homogeneously and inhomogeneously. The locations of the sensors are unknown. Specifically, we have three key contributions as follows: (i) We extend the distance distribution results of [68, 69] for Poisson Point Process and Binomial Point Process to the inhomogeneous deployment case (presented in Theorem 5.1); (ii)

We develop three different types of Askey-orthogonal polynomial expansion methods to approximate the marginal likelihood density (presented in Section 5.3). The first is based on Hermite polynomials and is known as the Gram-Charlier series expansion; the second is based on Laguerre polynomials and is known as the Gamma-Laguerre series expansion; the third expansion that we derive for the first time is the Beta-Jacobi series expansion and is based on Jacobi polynomials (presented in Theorem 5.2 and Theorem 5.3); (iii) We develop a novel analysis tool to characterise the conditions (skew-kurtosis region) at which each expansion has a positive support (presented in Section 5.3).

- In Chapter 6, we propose new distributed event algorithm to take into consideration incomplete information from sensors including Byzantine attacks and nonlinear transmissions. In section 6.1, we assume the channels between event and sensors are unknown. We develop a novel algorithm based on a statistical approach and model the Byzantine attack function in order to accurately detect the occurrence of events in the face of such attacks. We consider two practical cases where: 1) channels between events and sensors have unknown CSI; 2) sensors transmit to GW with unknown CSI. We also formulate an optimal Byzantine attack function from the attacker's point of view. We develop two distributed algorithms for these cases. We develop the optimal event detection decision rule under Byzantine attacks for the first case and a novel low-complexity event detection algorithm based on Gaussian approximation and Moment Matching for the second case which considers a global decision. In Section 6.2, we assume sensors have nonlinear transmission capabilities. We use the Beta-Jacobi series expansions (derived in Section 5.3) to approximate the likelihood under both hypothesis.
- Finally, Chapter 7 provides the conclusion and future research direction.

1.5 Publications

- (i) P. Zhang, I. Nevat, G. Peters, G. Xiao and H.-P. Tan, "Event detection in Wireless Sensor Networks with Random Censoring via Mixture Series Expansion," Submitted to IEEE Signal Processing Letter, August 2014.

- (ii) P. Zhang, I. Nevat, G. Peters, G. Xiao and H.-P. Tan, "Event Detection in Wireless Sensor Networks in Random Spatial Sensors Deployments," Submitted to IEEE Transactions on Signal Processing, July 2014.
- (iii) P. Zhang, G. Peters, I. Nevat, G. Xiao and H.-P. Tan, "Distributed Event Detection in Sensor Networks under Random Spatial Deployment," Accepted by Milcom 2014.
- (iv) P. Zhang, H.-P. Tan, and G. Xiao, "Distributed Algorithms for Maximizing Lifetime in Clustered Wireless Sensor Networks Using Energy-Harvesting Relay Nodes," Submitted to Computer Networks.
- (v) Zhang, Pengfei, Koh, Jing Yang; Lin, Shaowei, Nevat, Ido, "Distributed event detection under Byzantine attack in wireless sensor networks," ISSNIP 2014, pp.1,6, 21-24 April 2014.
- (vi) P. Zhang, G. Xiao, and H.-P. Tan, "Clustering Algorithms for Maximizing the Lifetime of Wireless Sensor Networks with Energy-Harvesting Sensors," Computer Networks, vol. 57, no. 14, pp. 2689-2704, Oct. 2013.
- (vii) Pengfei Zhang, Gaoxi Xiao and H. P. Tan. "Distributed Relay Scheduling for Maximizing Lifetime in Clustered Wireless Sensor Networks." Proc. of the IEEE ICCS, Singapore, pp 11-15, November 2012.
- (viii) P. Zhang, G. Xiao and H. Tan, "A Preliminary Study on Lifetime Maximization in Clustered Wireless Sensor Networks with Energy Harvesting Nodes." Proc. ICICS' 2011, Dec. 2011.

Chapter 2

Literature Review

In this thesis we study and develop solutions for various problems related to two fundamental aspects of sensor networks. These are

- Lifetime maximization of WSN.
- Event detection in WSN.

In this chapter we provide a comprehensive overview of existing works related to these two problems. In Table 2.1 we summarise the basic categories of each problem and the section in which they are discussed. Specifically, in Section 2.1, energy efficient networking algorithms for lifetime maximization are studied and in Section 2.2, distributed event detection algorithms are considered. In Section 2.3, background knowledge in statistical decision theory is covered. In Section 2.4, series expansion methods are covered.

2.1 Lifetime Maximization of WSN

2.1.1 Lifetime Definitions

In [2], the authors summarized different connectivity-based WSN lifetime definitions. The authors divided the connectivity-based network lifetime definitions into three classes, as shown in the Table 2.2. The coverage-based lifetime definition was considered in [80], where the network lifetime was defined as the time until a certain part of the network cannot be covered.

The most popular lifetime definitions belong to the third class. In [81], the network lifetime was defined as the time until the network became dysfunctional. According to different

Table 2.1: Categories of Algorithms

Categories	Section
Lifetime Maximization	
a. Clustering Battery Powered	2.1.2
b. Clustering Energy Harvesting	2.1.3
c. Non Clustering	2.1.4
Event Detection	
a. Deterministic Network	2.2.1
b. Random Network	2.2.2
c. Malicious Network	2.2.3

Table 2.2: Categorization of network lifetime definitions [2]

Class	Representative
1	Time until a predefined percentage of sensor nodes fail.
2	Time until the first partition in the network occurs.
3	Time until packet delivery rate falls below a predefined threshold.

applications, the definition varied, e.g., the time until the first node dies or the time until a certain fraction α of node dies [82]. In this thesis, we adopt this definition of network lifetime, and investigate the effects of different values of α . By adopting such a definition, maximizing network lifetime leads to a max-min problem with the objective of making those sensor(s) with shortest lifespan live as long as possible.

Energy efficient networking algorithms are classified according to clustering algorithms and non clustering algorithms. Also, both battery powered and energy harvesting WSNs are considered. In section 2.1.2, energy efficient clustering algorithms in battery powered WSN are considered. In section 2.1.3, energy efficient clustering algorithms in energy harvesting WSN are studied. In section 2.1.4, energy efficient non-clustering algorithms in WSNs are studied. In section 2.1.5, relay node placement in WSNs is studied.

2.1.2 Clustering Algorithms in Battery Powered WSN

Network lifetime maximization in battery powered WSNs can be roughly classified into four categories:

- Static clustering-based algorithms which did not change the clusters once they were formed up [27–29]. Such existing work, however, are based on different network topologies compared with ours, e.g., tree-based [27], chain-based [28], multiple-hop intra-cluster topologies [29], etc.
- Energy efficient static clustering algorithms which aimed at minimizing the transmission power of CHs, reducing the number of clusters [23–26, 83–85], load balancing [86, 87] and fault tolerance [88, 89]. These studies, however, mainly focused on (i) minimizing the number and/or the energy consumptions of CHs [23, 25, 26, 84, 85]; or (ii) efficiently utilizing mobile CHs [24].
- Dynamic clustering algorithms, which re-selected CHs periodically, adopting the same objective as, yet different network models from, ours. For example, a multi-level intra-cluster topology was studied in [30]. Some other works [31, 32] introduced heterogeneity into the network design, where CHs were equipped with more energy compared with NCHs; or introduced multiple (more than two) layers in the network topology, e.g., a super-CH layer to reduce the burden of CHs [33, 34].
- Dynamic clustering algorithms adopting the same network model and design objective as ours. They included: LEACH [13], TL-LEACH [90], EECS [15], PEGASIS [91], HEED [14], GROUP [92], MOECS [36], AEEC [39], and pLEACH [35]. Amongst these algorithms, LEACH was first proposed and widely served as a benchmark for comparison with other algorithms. MOECS was found to have longer lifetime compared with EECS and HEED [15, 36]. AEEC and pLEACH were recently proposed schemes that performed better than LEACH [35, 39]. However, no comparison between them and MOECS had been performed. LEACH [13], TL-LEACH [90], EECS [15], PEGASIS [91] and GROUP [92] typically selected the CHs rotatively with respect to their residual energy. The performances of these clustering algorithms in term of network lifetime were summarized in Table 2.3.

In this section, the problem of maximizing network lifetime in static clustering is considered. Since there are no existing algorithms designed for the same network model and design objective, our algorithm is extended to dynamic clustering for comparison. Specifically, the

Table 2.3: Network lifetime compared to standard LEACH [3]

Algorithm	Lifetime Characteristics	Reference
LEACH	First node death occurs 8 times later than the conventional methods, increasing network lifetime	[13]
TL-LEACH	Results in 30% improvement in lifetime versus the standard LEACH	[90]
EECS	35% improvement in network lifetime versus standard LEACH	[15]
PEGASIS	Simulations show a 100% to 300% increase in network lifetime versus the standard LEACH	[91]
GROUP	First node death occurs substantially later than that in the standard LEACH	[92]

extended algorithm is benchmarked with LEACH, MOECS, AECC and pLEACH. As will be shown in detail later in Section 3.5, the extended algorithm, though not optimized for dynamic clustering, manages to outperform all the existing methods.

Among all the clustering algorithms mentioned above, a few representative ones are selected for detailed discussions as below. Specifically, in section 2.1.2.1, the static clustering algorithms are studied. In section 2.1.2.2, the dynamic clustering algorithms are studied.

2.1.2.1 Static Clustering Algorithm

Ning et. al. [25] proposed an algorithm to achieve reliability and minimize the communication power for large networks. The proposed algorithm was more efficient and scalable than the result obtained by the Mixed Integer Nonlinear Programming (MINLP) solver. They also proposed to use the incremental deployment approach to solve the cluster head deployment problem. This new approach was more computationally expensive, yet provided a better solution than random approach.

In [23], the authors solved the MDEA (micro-server deployment and energy allocation) problem using MINLP solver. They studied the lifetime of two different configured networks, flat and hierarchical respectively, with the same total energy and quality of monitoring. They also proposed a heuristic approach to solve the MDEA problem. The approach was as follows: firstly they had a physical map of the sensors and constructed a graph, whose nodes were the

sensors with communication edges formed between them. Then they identified the Maximum Independent Set (MIS) of this graph such that each node in the MIS was the seed of micro-server cluster. Finally they performed node clustering until all nodes were assigned to clusters. They showed that the network lifetime was two or more times longer than a non-clustered WSN. The limitation was that the MINLP solver could not handle large-scale networks efficiently.

The authors in [26] proposed an incremental deployment algorithm to determine cluster head location, taking into account the multi-hop property. The purpose was to achieve minimal number of required sensors because they increase the cost of overall network. The idea was to achieve a minimal number of disjoint clusters by partitioning the network nodes, after which a CH was placed in each cluster. They adopted the Greedy Dominating Independent Set (DIS) heuristic to determine a set of cluster heads. They compared the performance of the proposed algorithm versus those of the augmenting placement and grid-based placement under capacity, cost, and QoS requirements and connectivity constraints. The results showed that the proposed algorithm required fewer number of cluster heads compared with the augmenting and grid-based placements. The algorithm however was not suitable for large-scale networks.

In [83], the authors proposed an uncapacitated facility location based clustering scheme. They added an additional layer of Super-Cluster-Head (SCH) nodes to extend the system lifetime. The scheme served to ease the CHs transmission load and balance the network node distribution. The SCH selection was formulated as an uncapacitated facility location problem (UFLP) and was solved to minimize the energy consumption during the CH-to-SCH-to-BS transmission process. The goal was to design a suitable hierarchy, which could reduce CHs transmission load, prolong system lifetime, and handle movable sensors. The drawback was that having one more layer increases system complexity and cost.

In [24], the authors proposed a technique including a weighted cost function based on the residual energy levels of cluster heads. This technique allowed the mobile actor to optimally fine-tune its geographical location. The main idea was that all the cluster heads should have a balanced energy dissipation rate, since a balanced energy dissipation rate helped achieve lower energy consumptions of CHs and enhanced the network lifetime.

In [82], the Linear Distance-based Scheduling (LDS) technique was proposed and analyzed. This technique could serve to decrease energy consumption with increased network

lifetime. Meanwhile, adequate sensing capabilities were maintained. In LDS, the sleeping probability of a sensor node was proportional to the distance from the cluster head. The energy consumption, the sensing coverage property, and the network lifetime of the proposed LDS scheme were analyzed. By comparing with the conventional Randomized Scheduling (RS) scheme with random sensor node sleeping probability, it was shown that the LDS scheme consumed 30% less energy than that of the RS scheme.

In [84], the author proposed an optimization algorithm for dividing the network into a smallest number of clusters by using merging method. Using the proposed algorithm, redundant cluster heads were eliminated, and unnecessarily overlapped clusters were merged. The proposed algorithm was compared with the Self Organizing Sensor algorithm(SOS) [93] and Algorithm for Cluster Establishment (ACE) [94]. Simulation results showed that the proposed algorithm achieved a lower number of clusters with better stability and shorter execution time.

In [85], the author proposed the Cluster Merging Algorithm (CMA). CMA merged small clusters to larger clusters with multi-hop intra-cluster communication through neighborhood discovery. The whole process included the following stages: A. Cluster head announcement; B. Cluster joining; C. Cluster merging; D. Link optimization; E. Gateway selection; F. Retirement of cluster heads and gateway; and G. Establishment of data gathering path. The performance of CMA was compared against that of TopDisc [95]. Simulation results showed that CMA achieved shorter topology establishment time and a lower network overhead.

In [86], the authors assumed that the network comprised normal energy constrained sensors and energy rich gateways, and the gateway nodes served as CHs. In the clustering phase, each gateway calculated the communication costs of the sensors within its range and shared the information with other gateways. Once clustering decisions had been made, the gateways shall inform every sensor which cluster it belongs to. The authors compared their results with the shortest distance algorithm and showed that the proposed algorithm achieved a smaller standard deviation per unit of load.

In [87], the authors studied the problem of task allocations of gateways (cluster heads) in order to maximize the lifetime of these cluster heads. Specifically, they formulated the task allocation problem into an optimization problem under several constraints. Then they adopted simulated annealing optimization method to solve this problem. Experimental results showed that optimized allocation of tasks always gave better results than the non-optimized case in term of energy usage.

2.1.2.2 Dynamic Clustering Algorithms

In [89], the author proposed LEACH which used randomized rotation of the cluster head to avoid rapid draining of the battery of any sensor in the network. [89] compared LEACH to LEACH-C (LEACH-Centralized), minimum-transmission-energy (MTE) routing, and static clustering in terms of system lifetime, energy dissipation, amount of data transfer, and latency. The time until the first node dies in LEACH was 8 times longer than that in direct transmission, MTE routing, and static clustering. The time until the last node dies was 3 times later than that in other protocols.

In [31], the authors proposed and evaluated the distributed energy-efficient clustering scheme (DEEC) for heterogeneous WSNs. In heterogeneous WSNs, sensor nodes were equipped with different amounts of energy. DEEC selected CHs using a probability based on the ratio between the residual energy of each node and the average energy of the network. The advantage of DEEC was that it did not require any global knowledge of energy at every election round. To make comparisons between DEEC and the other protocols including LEACH, Stable election protocol (SEP) [32], and Enhanced-LEACH (LEACH-E) [89], the author adopted various performance metrics such as the number of nodes alive, rounds until the first node dies, and rounds until 100% nodes die, etc. It was shown that the time until the first node dies was prolonged compared to that of the other protocols; and the network could have more rounds of operations as well.

A novel energy efficient clustering scheme (EECS) was proposed in [14], which better suit periodical data gathering applications. EECS was comprised of two steps, namely cluster head election and cluster formation respectively. In step one, every node with a probability of being selected as a cluster head, termed as candidate node, broadcasts within a certain range. If a more suitable node was found in the surrounding area, the candidate node would give up its right to be elected; otherwise, it would serve as the cluster head. In step two, every NCH node decided to join a CH according to the cost function. The cost function was defined by assigning weights to both the distance between CH and NCH and the distance between CH and BS. Simulation results showed that the proposed algorithm could achieve 130% longer lifetime than that of LEACH.

2.1.3 Clustering Algorithms in EHWSN

In this section, we review clustering algorithms in EHWSN (a) without EH relay in section 2.1.3.1 and (b) with EH relays in section 2.1.3.2.

2.1.3.1 Clustering Algorithms in EHWSN without EH relay

There have been many studies on EH-WSN [35, 44, 96–99]. The existing works on clustering algorithms in EH-WSN can be broadly classified into two categories: 1) a network was solely composed of EH sensors which have infinite lifetime [35, 96–98]. 2) a hybrid WSN which comprised of both battery-powered and EH nodes [44].

In [96], the author proposed to use solar power in wireless sensor networks and extended LEACH to be solar-aware. They proposed centralized and distributed algorithms. In the centralized solar-aware LEACH, CHs were selected in the same way as that in LEACH except that a higher probability was assigned to a solar-aware sensor with higher remaining energy to become CH. They compared the performance of solar-aware LEACH with standard LEACH under different environments. By adopting the performance metrics such as the time until the first node died, the authors showed that the proposed algorithm achieved 50% improvement compared to the standard LEACH.

In [97], two methods were proposed to improve environmental energy-based WSN network lifetime: the first method was E-EEHF (Enhanced-Environmental Energy Harvesting Framework); the second method was an optimized clustering method. E-EEHF was a more accurate and adaptive method than EEHF proposed in [41]. The main idea of the clustering method was to assign a low cost to a node with more residual energy in its rechargeable battery. They compared the performance of HEED (Hybrid Energy Efficient Approach [15])+EEHF, PC+EEHF with PC+E-EEHF in ns2, where PC denotes the proposed clustering. The network lifetime achieved its largest value while using PC+E-EEHF.

In [100], the authors proposed a partition-based centralized sLEACH (PC-sLEACH) which selected CHs based on current energy and energy harvesting rate of each node. The proposed partition-based centralized sLEACH (PC-sLEACH) consisted of two phases. The first phase was the partitioning phase. In this phase, the sink node collected the location and residual energy information from the sensors. Then the whole network was partitioned into K sectors by straight lines starting from the BS, where K is the predetermined number of clusters. A sector

was formed by two lines with central angle $2\pi/K$. In the case where sensors had a randomly uniform distribution, every sector had roughly the same number of sensors. The second phase was the CH election phase, where the sink node selected the node with the highest residual energy in each sector as the CH. All non-cluster head nodes joined the closest cluster head. The authors compared the performance between PC-sLEACH, D-sLEACH (Distributed-sLEACH), C-sLEACH (Centralized-sLEACH) [96] and A-sLEACH [44]. PC-sLEACH prolonged the network lifetime by about 20% - 25 % compared to other protocols.

In [44], the authors proposed the A-sLEACH: an advanced solar aware LEACH protocol, which was an extension of sLEACH in [96]. This approach adopted a new cluster head rotation method, a new radio energy model, and enhanced data aggregation by FIFO priority and Carrier Sense Multiple Access (CSMA) scheme. In the centralized A-sLEACH, CHs were selected by base station using the proposed scanning technique. Based on the assumptions that the base station was fixed and located far from the solar aware sensors, they compared the performance of A-sLEACH with sLEACH and showed that the network lifetime was improved by 19.58%. The proposed A-sLEACH was also better than sLEACH in other performance metrics such as energy dissipation per node, total remaining energy and number of sensors still alive over time. However, they let EH nodes serve as CHs with a higher probability than that of the battery-powered nodes.

2.1.3.2 Clustering Algorithms in EHWSN with EH relay

Since there are no existing works on studying CH-EH matching algorithms in wireless sensor networks, we look through the literature in other fields for possible references. Specifically, matching algorithms have been extensively studied with wide applications, e.g., those in optical VOQ networks [78] (refer to Figure 2.1). The representative algorithms include PIM (Parallel Iterative Matching)[78] and iSlip [79] that use random and round-robin approaches respectively for matching between input and output ports. Both algorithms are iterative and proven to converge in $\mathcal{O}(\log N)$ iterations on average.

We draw the following analogies between wireless sensor networks and optical VOQ networks:

- In wireless sensor networks, every node broadcasts to nodes within its range. In optical VOQ networks, input ports share the bandwidth to transmit to output ports.

Figure 2.1: Abstract model of optical VOQ switch [1].

- The distributed algorithm in optical VOQ networks requires fast convergence and low delay, which is also applicable in wireless sensor networks to provide low control overhead.
- The signaling scheme between input and output ports is analogous to the signaling between different nodes in wireless sensor network, in which coordination between nodes is required.

Based on the above similarities, we propose an EH-CH matching algorithm in section 4.2.

2.1.4 Non-Clustering Algorithms in WSN

There are also non-clustering energy efficient networking algorithms [4, 43, 80, 101, 102] in WSNs to improve various parameters of WSNs. These algorithms can be categorized into two category: 1) Algorithms focused on different lifetime definitions [80] (also discussed in Section 2.1.1) 2) Utilizing energy harvesting algorithms without clustering for lifetime maximization [4, 43, 101, 102].

In [80], the authors proposed a new Coverage Configuration Protocol (CCP) which, by using the proposed K_s -coverage eligibility algorithm, could dynamically configure a network

Table 2.4: Comparison of energy harvesting rates of various energy harvesting techniques [4]

Technology	Power Density [$\mu W/cm^2$]	Energy Harvest- ing Rate [mW]	Duty Cycle (%)
Vibration-electromagnetic	4.0	0.04	0.05
Vibration-piezoelectric	500	5	6
Vibration-electrostatic	3.8	0.038	0.05
Thermoelectric	60	0.6	0.72
Solar-direct sunlight	3700	37	45
Solar-indoor	3.2	0.032	0.04

to achieve guaranteed degrees of coverage and connectivity. In CCP, each node could have three states: SLEEP, ACTIVE, and LISTEN; and each node changed its state independently based on local information. The authors also tested different degrees of coverage that CCP could provide. Specifically, CCP was compared with Ottawa protocol [103]. The result showed that percentage of active number of nodes in CCP was less compared with Ottawa protocol. They also evaluated the capabilities of several other protocols including SPAN [104], CCP, SPAN+CCP, CCP-2Hop, SPAN+CCP-2Hop, etc. What was probably more important was that the paper also provided a geometric analysis on the relationship between coverage and connectivity. This analysis yielded some key insights for treating coverage and connectivity in a unified framework.

In [4], the authors performed a detailed review on the current state of art in energy harvesting wireless sensor networks. Table 2.4 showed a detailed comparison between various energy harvesting techniques. Amongst them, solar energy was the most promising and most widely used technique at this moment. The authors defined effective energy as a function of expected energy from a subsequent duration, energy consumed by non-optional task and current battery level. Effective energy was another important performance measurement metrics for EH-WSN.

In [101], the authors considered two parameters including the average energy harvesting rate and the variability of the source rate in the modelling of energy prediction. They used exponentially weighted moving-average (EWMA) filter to predict the future energy profile, based on which they formulated the problem into an optimization problem and derived a simple recursive algorithm to solve it. The objective of the optimization was to achieve maximal duty cycle subject to the constraint that energy neutral operation was satisfied.

In [102], the authors proposed duty cycle control mechanism for EH-WSN using adaptive control theory. Unlike previously proposed techniques [101] which were minimally adaptive and typically used a priori knowledge of energy profile, the author's approaches could achieve better performance in various energy profiles.

In [43], comparisons were made between several solar energy prediction algorithms, including Weather Conditioned Moving Average (WCMA) [105], Exponentially Weighted Moving Average (EWMA) [106], ETHZ algorithm [107] and neural network prediction [108]. The result showed that EWMA achieved average error less than 10%, which was much less than the errors by using other predictors.

2.1.5 Relay Node Placement in WSNs

In [109], the author considered relay node placement for two cases. In the case of one-tiered relay node placement, the algorithm can ensure that each pair of sensor nodes can be connected through either relay node or relay node/sensor node. In the case of two-tiered relay node placement, the algorithm can ensure that each pair of sensor nodes can be connected through relay nodes only. They considered different transmission ranges of relay nodes and sensor nodes. Algorithms for both cases are claimed to be polynomials but with approximate optimization.

In [110], the authors proposed a low complexity relay node placement algorithm in WSN. The objective of the work is to deploy the least number of relay nodes in a heterogeneous WSN to fulfil certain degree of connectivity. The heterogeneous WSN is defined as comprising nodes that may have different transmission radius. By providing the connectivity, the scheme can satisfy fault-redundant capabilities.

In [111], the author proposed relay node placement algorithm for WSNs in tunnels. The contributions of the papers are: (i) the author estimates the path loss distribution parameters and fading distribution models through a large number of experiments; (ii) they found that the channel models have significant effect on the number of relay nodes and placement of relay nodes; (iii) two tiered clustering based schemes are adopted; (iv) they optimized the placement of second tiered relay nodes through search and found algorithms.

In [112], the author studied the constrained relay node placement problem in WSN. They claimed that their work is the first work to study relay node placement in WSN where the relay nodes can only be placed at pre-determined locations. They also assumed the transmission

ranges of relay nodes are more than twice of the transmission range of sensor nodes. In terms of the algorithm, they considered two problems. In the first problem, they aimed to find the minimum number of relay nodes in order to make sure each sensor is connected to at least one relay node or BS. Relay nodes and BS form the connected network. In the second, they aimed to find the minimum number of relay nodes in order to make sure each sensor is connected to at least two relay nodes or BS, resulting in a 2-connected network.

2.2 Event detection in Wireless Sensor Networks

Distributed event detection algorithms in WSNs can be categorized into three categories:

- Deterministic WSNs models [8, 49–52, 55–59]. These algorithms assume that the GW has perfect knowledge of the locations of the sensors and of the wireless channels.
- Random WSNs models [54, 62, 65, 68–72, 113]. In these models the deployment of sensors were based on spatial point process and their locations are considered random unknown to the GW. At the center of these models is the random distance between the event and the sensors. As such, different approaches have been developed to derive the distribution of random distances. Distance distribution in random networks and possible applications were discussed in [68, 69, 113]. Discussions on SINR and possible applications in different networks were preformed in [65]. A survey of stochastic geometry in WSN can be found in [70]. Interference and outage in mobile random networks is covered in [71]. Discussions about the density of random WSN to achieve certain event detection probability were done in [72].
- WSNs under malicious attacks [9, 10, 74, 75, 114, 115]. In this thesis, we consider a particular kind of attack on the network from the malicious users participating in the network itself. This attack is termed as a Byzantine attack in the literature and the attackers are called Byzantine attackers. Byzantine attackers transform their observations (or some summary statistics of their observations). The aim of the Byzantine attackers is to make the GW detect an incorrect state of nature and then utilize this phenomenon to achieve certain advantages. Byzantine attack problem in event detection for WSN have been discussed in [9, 10, 75]. A more general outlier or anomaly detection problem was considered in [74, 114, 115].

These existing algorithms are discussed in details in the following sections. Specifically, in Section 2.2.1, distributed event detection algorithm in deterministic WSNs is discussed. In Section 2.2.2, distributed event detection algorithms in random WSNs are studied. Section 2.2.3 focused on event detection algorithms in WSNs with malicious users.

2.2.1 Distributed Event Detection Algorithms in Deterministic WSNs

In [59] the problem of distributed detection was considered, where the sensors transmit their local decisions over perfectly known wireless channels. Specifically, the authors proposed a distributed detection system infrastructure with a virtual multi-input multi-output (MIMO) channel to account for non-ideal communications between a finite number of sensors and a FC (fusion center). Each sensor has an individual transmission power constraint, and a joint power constraint on the the total amount of power that the sensors can spend to transmit their local decisions to the FC was imposed.

Theoretical performance analysis was derived in [57] for detection fusion under conditionally dependent and independent local decisions. Specifically, the paper was related to the maximum ratio combining (MRC) fusion rule over MIMO channel. They obtained the closed form expressions of the instantaneous-channel (IC) system probabilities of false alarm and detection. They also derived the closed form expression of the channel-averaged (CA) conditional moment generating function (MGF) of the statistics. Also, the CA conditional MGF was exploited to show the large-system limit under both individual power constraint (IPC) and total power constraint (TPC) scenarios, leading to a non-ideal receiver operating characteristic (ROC) in both cases.

Distributed detection in sensor networks over fading channels with multiple receive antennas at the GW was considered in [58]. They derived the optimal decision rules and the associated probabilities of detection and false alarm for three scenarios of Channel State Information (CSI) availability. For the most difficult case of unknown CSI, they developed two new algorithms to derive the optimal decision rule. The first was based on a Gaussian approximation method where they quantified the approximation error and its rate of convergence (to a true Normal distribution) via a multivariate version of the Berry-Esseen bound. The second was based on a multivariate Saddle-point (Laplace) approximation which was obtained via

a non-convex optimization problem which was solved efficiently via Bayesian Expectation-Maximization method.

In [55], the author obtained an optimal detection procedure that minimized the detection delay between the change time and the stopping time while the probability of false alarm did not exceed a desired quantity. Furthermore, they also obtained an optimal sampling rate which minimized the mean detection delay.

In [56], the author considered the problem of sensor selection for event detection in WSNs. They wanted to choose a subset of p out of n sensors that yielded the best detection performance. They proposed two sensor selection optimality criteria: 1) the Kullback-Leibler (KL) distance; and 2) the Chernoff (C) distance between π_1 and π_0 . $\pi_i, i = 0, 1$ are p dimensional Gaussian distribution from each possible p -sensor selection under hypothesis H_i . They proved that the KL and C sensor selection problems are NP hard and it was unlikely to find an efficient algorithm that solved large instances of these problems. They proposed a computationally affordable algorithm and demonstrated the algorithm's near optimality in a very large portion of problem instances.

In [50], the author constructed a universal decentralized scheme for the detection of a deterministic signal corrupted by additive noise with unknown distribution by using the isotropic decentralized estimation scheme. Operationally, each sensor collected an observation, performed a local probabilistic compression, and sent a 1-bit message to the fusion center, while the latter performed the final detection by appropriately combining the received bits from all sensors.

In [52], the author combined the decisions made by individual sensor nodes to achieve system detection performance beyond a weighted average of individual sensor nodes. They proposed a centralized threshold-OR fusion rule for combining the individual sensor node decisions in the sensor network. They derived threshold bounds for accumulated decisions using Chebyshev's inequality based on individual hit and false alarm probabilities but without requiring a priori knowledge of the underlying probability densities. They analytically showed that the fused method achieved a higher hit rate and lower false alarm rate compared to the weighted averages of individual sensor nodes.

2.2.2 Distributed Event Detection Algorithms in Random WSNs

Next, we turn our attention to scenarios where the sensors are randomly deployed in the field and the FC has no a-priori knowledge regarding their locations. We discuss the properties of such event detection algorithms in Section 2.2.2.1. Then, distributed event detection algorithms in random WSNs are reviewed in Section 2.2.2.2.

2.2.2.1 Properties of Random WSNs

In [62], the author focused on the estimation of pseudo-random deployment models that exploited different amounts of a priori information. Additionally, they considered general sets of interesting points. Topologies derived from the models required minimal energy consumption at maximum sensing coverage. Furthermore, they proposed a novel metric enabling the performance estimation of WSNs under different deployment strategies. This metric showed a direct relationship to their performance metric applied in the evaluation of the estimated models. They considered a scenario where deployment had to be accomplished in one step and no intermediate information became available during deployment. The aim of their approach was the optimization of initial deployments which could subsequently be used by other existing self-organization methods, to further improve the deployment through mobility-based techniques. The proposed scheme could lead to the minimization of the total moving distance, number of moves, communication and computation cost, and convergence rate of those methods.

In [68], sensors were uniformly randomly distributed following a Binomial Point Process. The authors analyzed various properties of such networks including the distance distribution, moments of distance etc.

In [69], the authors consider a more general distribution, namely the Poisson Point Process (PPP) and provide analysis of the distance distributions for such networks.

In [70] the authors discussed the deployment of cognitive cellular wireless networks. They focused on the related literature on stochastic geometry modeling and analysis of single-tier, multi-tier, and cognitive cellular networks. They also provided a detailed taxonomy of the existing literature according to the point process used, the interference modeling approach, and the target network model. The proposed taxonomy revealed the popularity and applicability of the different point processes (PPs) and modeling techniques.

2.2.2.2 Distributed Event Detection Algorithm in Random WSNs

In [54], they proposed a new decision fusion rule utilized for a random WSN where the number of sensors was assumed to be a Poisson random variable and the sensors locations were unknown to the FC. The fusion rule was based on the Bayes Factor whose performance was shown to be better than that of the Counting Rule and the Scan Statistic; they also highlighted the difference between the Scan Statistic test and the optimal NP test.

In [72], the authors seek to provide guidelines on how dense a sensor system should be for the scenario where correlation among observations increased with sensor node density. Specifically, they considered a scenario where the sensor nodes were placed on a line, and the observations at the sensors were samples of a spatial stochastic process on the line. They assumed that the nodes were subject to a power density constraint, i.e., the power per unit distance was fixed, so that the power per node decreased linearly with node density. They then employed mathematical instruments from large deviation theory and statistical signal processing to derive guidelines relating sensor density, resource allocation, and overall system performance.

2.2.3 Distributed Event Detection Algorithms in WSNs with Malicious Users

A popular approach to identifying Byzantine attack involved local decisions of sensors [9]. They proposed a simple and efficient reputation-based detection scheme to identify Byzantine attackers by counting mismatches between their local decisions and the global decision at the FC over a time window and then removed the Byzantines from the data fusion process. The proposed scheme was robust against Byzantine attacks and it could successfully removing the Byzantines in a very short time span.

In [10], the authors considered the problem of joint optimization of the fusion rule and local sensor thresholds for a fixed Byzantine strategy. They considered the problem of designing optimal distributed detection parameters in a holistic manner in the presence of Byzantines and also modeled the strategic behavior of the FC and the Byzantines using game theory. They analyzed the problem under different attacking scenarios and gave insights into the practical scenarios where the proposed schemes were useful. The main contributions of this paper were: 1). the problem of joint optimization of the fusion rule and local sensor threshold was considered for a fixed Byzantine strategy; 2). they extended these results to the scenario where both

the FC and the Byzantine attacker acted in a strategic manner to optimize their own utilities and modeled it using Game Theory; 3). numerical results were provided to gain insights into the solution.

Other methods to detecting malicious users in wireless sensor networks were also considered. In [114], a witness-based approach is used to determine the validity of results. They proposed to use the witness concept to solve the assurance problem between data fusion nodes and the base station. Some nodes around the data fusion node were selected as witnesses to monitor the data fusion results. The proposed method assured the validity of data fusion results received by the base station.

In [74], a detailed survey was given on the problem of identifying anomalous data. The author attempted to provide a structured and a broad overview of extensive research on anomaly detection techniques spanning multiple research areas and application domains. They discussed various techniques and identified unique assumptions regarding the nature of anomalies made by the techniques. They distinguished the simple anomalies from complex anomalies.

In [115], the anomalous data was identified using Bayesian Belief Networks. They proposed a technique for the outlier detection and estimated missing values using Bayesian Belief Networks (BBN) in the sensor stream data. Apart from identifying the outliers, they could also estimate the missing values in the sensor network. The proposed scheme went through the following phases: training phase, testing phase, and inference phase. Training phase trained the BBN to capture the spatio-temporal correlations among the sensor nodes. In order to test the level of accuracy the network has learned, testing phase was used and the learned parameters were updated if needed. Inference phase inferred the missing values and detected whether the stream data generated was an outlier or not.

Our approach is different from previous related works in the following ways:

- (i) Instead of using local decisions by each sensor, each sensor transmits its raw information to the GW and the GW makes a global optimal decision based on the raw information.
- (ii) Instead of assuming perfect CSI for the channels, we consider the case when CSI is random and unknown.
- (iii) Previous works do not deal with malicious users in the context of event detection. In this thesis, we optimize the Byzantine attack function, and propose optimal decision rules to counter such attacks.

Next we present a shot introduction to statical decision theory and series expansion methods which we adopt to solve the event detection problems.

2.3 Statistical Decision Theory

In our daily life, we are constantly making decisions. Given some hypothesis, a criterion is selected, upon which a decision has to be made. The process that the receiver undertakes in selecting a criterion falls under the theory of signal detection.

The problem can be described by a source emitting two possible outputs at various instants of time. The output are referred to as hypothesis. The *null hypothesis* $\mathcal{H}_0$ represents a zero (target not present) while the *alternative hypothesis* $\mathcal{H}_1$ represents a one (target present). Assume Y denotes the random variable which is observed from receiver. Based on the observation values of Y , the receiver decides which hypothesis ($\mathcal{H}_0$ or $\mathcal{H}_1$) is true.

Let $(\Theta, \Delta, \mathcal{L})$ be a statistical game with $\Delta = (\delta_0, \delta_1)$. We observe a random variable Y taking values in a space $\mathcal{Y}$. The probability density of Y corresponding to each hypothesis are $f_{Y|\mathcal{H}_0}(y|\mathcal{H}_0)$ and $f_{Y|\mathcal{H}_1}(y|\mathcal{H}_1)$. $\mathcal{H}_0$ and $\mathcal{H}_1$ are depending on θ which lies in a parameter space Θ . We desire to fashion a *decision rule* $\phi : \mathcal{Y} \rightarrow \mathcal{R}$ such that, when $Y = y$ is observed,

$$\phi(y) = \begin{cases} 1, & \text{if } y \in \mathcal{R} \\ 0, & \text{if } y \in \mathcal{A} \end{cases}$$

$$\begin{aligned} \mathcal{H}_0, & \quad \theta \in \Theta_0 \\ \mathcal{H}_1, & \quad \theta \in \Theta_1 \end{aligned}$$

where $\mathcal{R}$ and $\mathcal{A}$ are measurable subsets of $\mathcal{Y}$, and $\mathcal{Y} = \mathcal{R} \cup \mathcal{A}$.

Each time a decision is made, based on some criterion, for this binary hypothesis testing problem, four possible cases can occur:

- (i) Decide $\mathcal{H}_0$ when $\mathcal{H}_0$ is true.
- (ii) Decide $\mathcal{H}_0$ when $\mathcal{H}_1$ is true.
- (iii) Decide $\mathcal{H}_1$ when $\mathcal{H}_0$ is true.
- (iv) Decide $\mathcal{H}_1$ when $\mathcal{H}_1$ is true.

For cases (i) and (iv), the receiver makes a correct decision (P_D), while for cases (ii) and (iii) the receiver makes an error. Case (2) causes miss detection (P_M) and case (3) is false alarm (P_F).

The ratio of $f_{Y|\mathcal{H}_1}(y|\mathcal{H}_1)$ over $f_{Y|\mathcal{H}_0}(y|\mathcal{H}_0)$ is called the *likelihood ratio* and denoted by $\Lambda(y)$. The optimal decision rule is a threshold test based on the likelihood ratio [116].

$$\Lambda(y) = \frac{f_{Y|\mathcal{H}_1}(y|\mathcal{H}_1)}{f_{Y|\mathcal{H}_0}(y|\mathcal{H}_0)} \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\gtrless}} \gamma$$

P_M and P_F can be expressed below:

$$P_M = \int_{Z_0} f_{Y|\mathcal{H}_1}(y|\mathcal{H}_1) dy$$

$$P_F = \int_{Z_1} f_{Y|\mathcal{H}_0}(y|\mathcal{H}_0) dy$$

$$P_D = \int_{Z_1} f_{Y|\mathcal{H}_1}(y|\mathcal{H}_1) dy$$

Z_0 and Z_1 corresponding to the regions where null and alternative hypothesis happen.

Several criteria are used to choose γ in the likelihood ratio test. For example, Neyman-Pearson criterion chooses γ such that P_F is fixed to some value α and P_D is maximized. Mathematically, it is to minimize the objective function J where

$$J = \lambda(1 - \alpha) + \int_{Z_0} [f_{Y|\mathcal{H}_1}(y|\mathcal{H}_1) - \lambda f_{Y|\mathcal{H}_0}(y|\mathcal{H}_0)] dy$$

$\lambda(\lambda \geq 0)$ is the *Lagrange multiplier*. The decision rule is, therefore

$$\Lambda(y) = \frac{f_{Y|\mathcal{H}_1}(y|\mathcal{H}_1)}{f_{Y|\mathcal{H}_0}(y|\mathcal{H}_0)} \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\gtrless}} \lambda$$

This criterion is based on Neyman-Pearson Lemma. Suppose that $\Theta = \{\theta_0, \theta_1\}$ and that the distributions of Y have densities $f_Y(y|\theta)$. Any test $\phi(Y)$ of the form

$$\phi(y) = \begin{cases} 1, & \text{if } f_Y(y|\theta_1) > k f_Y(y|\theta_0) \\ \gamma(y), & \text{if } f_Y(y|\theta_1) = k f_Y(y|\theta_0) \\ 0, & \text{if } f_Y(y|\theta_1) < k f_Y(y|\theta_0) \end{cases}$$

for some $k \geq 0$ and $0 \leq \gamma(y) \leq 1$, is best of its size for testing $\mathcal{H}_0, \theta = \theta_0$ against $\mathcal{H}_1 : \theta = \theta_1$.

Another criterion to choose γ is Bayes' criterion. The decision rule resulting from the Bayes' criterion is

$$\Lambda(y) = \frac{f_{Y|\mathcal{H}_1}(y|\mathcal{H}_1)}{f_{Y|\mathcal{H}_0}(y|\mathcal{H}_0)} \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\gtrless}} \frac{P_0(C_{10} - C_{00})}{P_1(C_{01} - C_{11})}$$

P_0 and P_1 are the prior probability of $\mathcal{H}_0$ and $\mathcal{H}_1$ respectively. C_{ij} is the cost of deciding $\mathcal{H}_i$ given the true hypothesis is $\mathcal{H}_j$.

A plot of P_D , versus P_F , with the threshold γ is referred to as receiver operating characteristic (ROC) curve. The ROC depends on the conditional density function of the observed signal under each hypothesis, that is, $f_{Y|H_j}(y|H_j), j = 0, 1$. Varying the values of γ , the values of P_D and P_F vary.

The accuracy of a binary hypothesis testing results is commonly summarized by the false alarm and the positive detection probabilities. When the test results are not simply under null or alternative hypotheses, but are measured on continuous or ordinal scales, the ROC curve is the most popular approach for assessing the accuracy. The ROC curve is defined as a plot of the positive detection probability against false alarm probability across all possible choices of threshold values. Each point of the ROC curve is generated by a different decision threshold. The ROC curve is a global assessment of the accuracy of a binary hypothesis problem and it does not require selection of a particular decision threshold since all possible decision thresholds are included.

The remaining tasks are to determine $f_{Y|H_j}(y|H_j), j = 0, 1$ which is usually mathematical intractable. Next we introduce the series expansion methods to approximate $f_{Y|H_j}(y|H_j), j = 0, 1$.

2.4 Series Expansions Methods

Calculating the PDF of a random variable is one of the most fundamental tasks in any inference procedure. In many of these cases, the PDF is intractable, and cannot be expressed in a parametric form, for example as a Gaussian distribution. In these cases, it is necessary to develop low complexity techniques to approximate the PDF of such random variables. One of such

approximation methods is series expansion method. This type of series expansion method assumes an existence of a reference PDF which almost always has a known analytic form. Then the approximated PDF can be expressed as a function of the reference PDF and polynomials based on moments and cumulants. We explain in details as below [117]:

Given a random variable X with support from $-\infty$ to $+\infty$. $f(x)$ is the probability density function (PDF) of x , $f(x)$ has the following property:

$$\int_{-\infty}^{+\infty} f(x)dx = 1.$$

Some important properties related to $f(x)$ include the moments μ_j and cumulants κ_j .

$$\mu_j = \mathbb{E}_f[x^j] = \int x^j f(x)dx.$$

Cumulants κ_j is related to μ_j with the following equations:

$$\kappa_1 = \mu_1$$

$$\kappa_2 = \mu_2 - \mu_1^2$$

$$\kappa_3 = \mu_3 - 3\mu_1\mu_2 + 3\mu_1^3$$

$$\kappa_4 = \mu_4 - 4\mu_1\mu_3 - 3\mu_2^2 + 12\mu_1^3\mu_2 - 6\mu_1^4$$

$f(x)$ can be approximated by the following series expansions:

$$f(x) \approx c_0\phi_0(x) + c_1\phi_1(x) + \cdots + c_l\phi_l(x) + \cdots$$

$\phi(x)$ is a given orthonormal set. c_l is called the Fourier coefficients of $f(x)$ defined by:

$$c_l = \int_a^b f(x)\phi_l(x)w(x)dx$$

ϕ_n and ϕ_m is orthonormal with respect to a weight function $w(x)$ if

$$(\phi_n, \phi_m) = \int_a^b \phi_n(x)\phi_m(x)w(x)dx = \delta_{nm}, n, m = 0, 1, 2, \cdots, l$$

Here $\phi_n(x)$ is real-valued and belongs to the class $\mathcal{L}^2(a, b)$.

Let the real-valued functions $f_0(x), f_1(x), \cdots, f_l(x)$ be of the class $\mathcal{L}^2(a, b)$ and linearly independent. Then an orthonormal set of functions $\phi_0(x), \phi_1(x), \phi_2(x), \cdots, \phi_l(x)$ exists such that, for $j = 0, 1, 2, \cdots, l$.

$$\phi_j(x) = c_{j0}f_0(x) + c_{j1}f_1(x) + \cdots + c_{jj}f_j(x), c_{jj} > 0$$

The orthonormal set is uniquely determined.

Let $\phi_j(x)$, $f(x)$, c_j have the same meanings as in the above definitions. Let $l \geq 0$ be a fixed integer, and $a_0, a_1, \dots, a_l$ arbitrary real constants. If we write

$$g(x) = a_0\phi_0(x) + a_1\phi_1(x) + \dots + a_l\phi_l(x)$$

and the coefficients a_j are variable, the integral

$$\int_a^b \{f(x) - g(x)\}^2 w(x) dx$$

becomes a minimum if and only if $a_j = c_j, j = 0, 1, 2, \dots, l$

With extension to orthogonormal polynomials P_j where $P_j = x^j + \text{lower order terms}$, $j = 0, 1, \dots$ and a sequence of positive numbers $\{\varepsilon_n\}$, with $\varepsilon_0 = 1$ such that:

$$\int_{\mathbb{R}} P_m(x) P_n(x) w(x) dx = \delta_n \delta_{m,n}$$

So in general, $f(x)$ can be approximated by $w(x)$, $P_m(x)$ and coefficients C shown below:

$$g(x) \approx w(x) \sum_{m=0}^{+\infty} C_m P_m(x)$$

An example to show Edgeworth series expansion is given in Fig. 2.2. The exact distribution, which follows a gamma distribution: $\bar{X} \sim \text{Gamma}(\alpha = 3, \theta = 2/3)$. The definition of edgeworth series expansions is give as:

$$g(x) = \phi\left(\frac{x - \mu}{\sigma}\right) \left(1 + \frac{\kappa_3}{6\sigma^3} H_3\left(\frac{x - \mu}{\sigma}\right) + \frac{\kappa_4}{24\sigma^4} H_4\left(\frac{x - \mu}{\sigma}\right) + \frac{\kappa_3^2}{72\sigma^6} H_6\left(\frac{x - \mu}{\sigma}\right)\right)$$

where $\phi(x)$ is the standard normal distribution and H_i is the Hermite polynomials which is also defined in section 5.3. The figure shows an accurate estimation of PDF compared with true PDF.

In Chapter 5, we discuss in details three different series expansion methods. Specifically, $w(x)$ are normal basis, gamma basis and beta basis. The corresponding polynomials $P_m(x)$ is Hermite polynomials, Laguerre polynomials and Jacobi polynomials respectively.

Figure 2.2: Density of true PDF and approximated PDF using Edgeworth Series Expansions.

Chapter 3

Centralized Lifetime Maximization Clustering Algorithms for WSNs

This chapter presents several efficient clustering algorithms for both single- and multi-clustered networks, with and without EH relay sensors respectively, with the objective of maximizing the network lifetime. Specifically, Section 3.1 shows the system model for centralized clustering algorithms. Section 3.2 shows the clustering algorithms without EH relay sensors. In Section 3.3, we extend the algorithms to include energy harvesting relay nodes. In Section 3.4, we discuss variants of clustering algorithms for different definitions of network lifetime. In Section 3.5, extensive simulation results and discussions are presented for verifying the performance of the proposed algorithms and evaluating the effects of EH sensors on prolonging network lifetime. Section 3.6 concludes this chapter.

3.1 System Model

Throughout this chapter, without loss of generality, we assume that N_s sensor nodes with known location and the same amount of initial energy are deployed in 2-D space, and a single BS with infinite energy is located at (0,0). The N_s sensor nodes will be partitioned into N_c clusters, each comprising one CH. We also assume TDMA-based communication is adopted for every cluster where each TDMA frame comprises N_s slots: the NCH nodes will transmit to the respective CH nodes in the first $N_s - N_c$ slots; the CH nodes then forward the received data to the BS in the subsequent N_c slots. As in [13], we consider the Friis free-space propagation model where the transmission power is proportional to the square of the distance. Note that

Table 3.1: Notations used throughout this chapter

Notation	Description	Value
N_s	Number of sensors in the network	$\{100, 150, 200\}$
N_c	Number of clusters in the network	$\{1, 2, \dots, 10\}$
$N_{s,j}$	Number of sensor nodes in cluster j	N.A.
i	Node index	$\{i=1, 2, \dots, N_s\}$
j	Cluster index	$\{j=1, 2, \dots, N_c\}$
k	Index of iteration	N.A.
r	Data transmission rate in bits/s	2.5×10^5 bits/s
E_{elec}	Energy required for processing each bit of data	5×10^{-8} J/bit
E_{DA}	Energy consumption for data aggregation	5×10^{-9} J/bit/signal
E_{amp}	Coefficient of energy consumption by transmission amplifier	10^{-10} J/bit/m ²
E_s	Energy stored in battery for each sensor	0.5J
CH_j	The CH node for cluster j	N.A.
$NCH_{f,j}$	The NCH node that is farthest away from the CH_j in cluster j	N.A.
(x_i, y_i)	Coordinates for node i	N.A.
$d_{A,B}(k)$	Distance between nodes A and B in iteration k	N.A.
t	Lifetime for CH	N.A.
$P_{EH_j,h}$	Harvesting rate for EH node in cluster j	N.A.
$P_{EH_j,c}$	Energy consumption rate for EH node in cluster j	N.A.
$P_{CH,off}$	Energy consumption rate for CH when EH is not working	N.A.
$P_{CH,on}$	Energy consumption rate for CH when EH is working	N.A.
t_{EH}	Time duration when EH node works	N.A.

the proposed algorithms can be easily extended to other propagation models, e.g., multi-path fading model [118]. The notations used in the chapter are shown in Table 3.1.

3.2 Lifetime Maximization Clustering Algorithms

Specifically, in Section 3.2.1, we propose an algorithm for finding the optimal location of the CH in a single-clustered WSN; this is then extended to multi-cluster networks in Section 3.2.2. In Section 3.2.3, we extend the clustering algorithms to dynamic clustering cases.

3.2.1 Cluster Head Selection Algorithm ($N_c = 1, \alpha = \frac{1}{N_s}$)

We describe our proposed cluster head selection algorithm for a single-cluster network. Amongst the N_s nodes, one node will be selected as the CH, which is assumed to be able to fuse information from all NCHs. Our single-cluster network model is depicted in Figure 3.1 and the objective of the algorithm design is to find the optimal location of CH for maximizing the time until the first node dies (i.e., $\alpha = \frac{1}{N_s}$).

Figure 3.1: The single-cluster WSN model.

Denoting by NCH_f the NCH node that is farthest away from the CH, and the distance between nodes A and B by $d_{A,B}$, the power required for CH and NCH_f is shown below:

$$P_{CH} = E_{elec}r(N_s - 1) + E_{DA}rN_s + E_{elec}r + E_{amp}d_{CH,BS}^2r \quad (3.1)$$

$$P_{NCH_f} = E_{elec}r + E_{amp}d_{CH,NCH_f}^2r \quad (3.2)$$

Intuitively, we know that either the CH or the NCH_f will die first since they consume more energy than any other NCHs. Hence the problem is to find the optimal location for the CH, (x^*, y^*) , which minimizes $\max(P_{CH}, P_{NCH_f})$. It is easy to see that at the optimal location, we shall have $P_{CH} = P_{NCH_f}$.

This problem is closely related to the well-known *weighted smallest circle problem* [119], which is defined as follows: denote the location of node i as $p_i = (x_i, y_i)$ and its positive weight as $w_i, i = 1, \dots, n$. For any point $p = (x, y)$, let d_{p,p_i} be the distance between p and p_i and

$$H(p) = \max_{1 \leq i \leq n} w_i d_{p,p_i}.$$

Find $p^* = (x^*, y^*)$ such that H is minimized.

To tackle the weighted smallest circle problem, Hearn and Vijay proposed a simple yet efficient optimization algorithm [119]. The main idea is to start with a minimum circle covering at least two nodes and then iteratively find among all the nodes outside the circle (if any) the one weighted-farthest away from the circle center, and update the circle to cover it. In each iteration, it is ensured that, for those nodes on the boundary of the existing circle and the weighted-farthest away node, the newly built circle is the minimum one which covers all these nodes by going through at least two or three of them on the boundary. It can be proved that the algorithm converges to the optimal solution.

By ignoring the item $E_{elec}r$ and $E_{elec}r(N_s - 1) + E_{DA}rN_s + E_{elec}r$, which are constants, in (3.1) and (3.2) respectively, we have that the energy consumption of each NCH is proportional to the square of its distance to CH and the energy consumption of CH is largely decided by the distance from BS to CH. Finding the best location of CH is equivalent to finding p^* for all the NCHs and the BS. If (and only if) the weight of each NCH and the BS is properly assigned such that when H in the minimum weighted circle problem is minimized, $\max(P_{CH}, P_{NCH_f})$ is also minimized.

Since the energy consumption of each NCH (minus a constant) is proportional to its distance to CH, we conveniently let $w_i = 1$ for $i = 1, 2, \dots, N_s$. For BS, denote its weight as w_0 . To take into account the energy consumption by CH for data processing and aggregation, we shall have

$$w_0 = \sqrt{\frac{E_{elec}(N_s - 1) + E_{DA}N_s}{E_{amp}d_{CH,BS}^2} + 1} \quad (3.3)$$

By doing so, when $w_0d_{CH,BS} = d_{CH,NCH_f}$, we have $P_{CH} = P_{NCH_f}$.

The value of d_{CH,NCH_f} remains unknown until the optimal solution of p^* is found, which prohibits finding the best solution of w_0 at the beginning. In our algorithm, we initialize $w_0 = 1$ and update its value iteratively. As we will see later, the iterative calculations converge.

The proposed algorithm is presented in detail as Algorithm 3.1. For simplicity of notation, we let $w(k)$ denote the weight of BS (i.e., w_0 above) in the k^{th} iteration, and $d(k)$ the distance between NCH_f and CH in the k^{th} iteration. Note that only w_0 needs to be updated in each iteration, while for all the other sensors, their weights remain as 1 throughout the calculation.

It can be proved that the algorithm converges.

Algorithm 3.1 Cluster Head Selection Algorithm

- **Initialization**

Input set of sensor locations $A = (x_1, y_1), (x_2, y_2) \dots (x_{N_s}, y_{N_s})$. Let $(x_0, y_0) = (0, 0)$ be the coordinates for the BS, and set $w(1) = 1$.

- **Step 1**

Let $k = 1$. Run the weighted smallest circle algorithm [119] to find the center of the circle, record the coordinate as $(x(1), y(1))$, and record $d(1)$.

- **Step 2**

Let $k = k + 1$. Set $w(k)$ according to (3.3) and run the weighted smallest circle algorithm to find the center for the k^{th} iteration. Record the coordinate as $(x(k), y(k))$; record $d(k)$ and update $d_{CH,BS}(k)$.

- **Step 3**

If $|d(k) - d(k - 1)| \leq \epsilon$, go to step 4; otherwise, go to step 2.

- **Step 4**

Having obtained the optimal CH position, (x^*, y^*) , we select the nearest sensor as CH.

Lemma 3.1. *Given $w(k + 1) \geq w(k)$, we have $d(k + 1) \geq d(k)$.*

Proof. We prove by contradiction. Given $w(k + 1) \geq w(k)$, let's assume that $d(k + 1) < d(k)$. Let $(\bar{x}(k), \bar{y}(k)) = (x(k + 1), y(k + 1))$, and $\bar{d} = d(k + 1)$. Then we have a feasible solution for the weighted smallest circle problem in the k^{th} iteration, $\bar{d} = d(k + 1) < d(k)$, which leads to a contradiction. $\square$

Theorem 3.1. *If $w(k + 1) \geq w(k)$, then either the algorithm terminates, i.e., $d(k + 1) = d(k)$, or $d_{CH,BS}(k + 1) \leq d_{CH,BS}(k)$.*

Proof. Let $(\bar{x}(k + 1), \bar{y}(k + 1)) = (x(k), y(k))$ and $\bar{d} = \max\{w(k + 1)d_{CH,BS}(k), d(k)\}$. We have a feasible solution for the $(k + 1)^{th}$ iteration where the center of the circle is at $(x(k), y(k))$ and the radius is $\bar{d}$.

If $\bar{d} = d(k)$, then $d(k + 1) \leq \bar{d} = d(k)$. From Lemma 3.1, $d(k + 1) = d(k)$, and the algorithm terminates.

If $\bar{d} = w(k + 1)d_{CH,BS}(k)$, we have $w(k + 1)d_{CH,BS}(k + 1) \leq d(k + 1) \leq \bar{d} = w(k + 1)d_{CH,BS}(k)$, and hence $d_{CH,BS}(k + 1) \leq d_{CH,BS}(k)$. $\square$

Theorem 3.2. $d(k)$ will converge within finite steps of k .

Proof. From (3.3), we have $w(2) > w(1)$. By Theorem 3.1, we have $d_{CH,BS}(2) \leq d_{CH,BS}(1)$. According to (3.3), $w(k+1)$ increases with a decreasing value of $d_{CH,BS}(k)$. Consequently we have the conclusion that $w(k)$ is a monotonically increasing function of k , while $d_{CH,BS}(k)$ is a monotonically decreasing function of k . Since $d_{CH,BS}(k) \geq 0$, $d_{CH,BS}(k)$ is a lower-bounded monotonic sequence, which definitely converges. Since $w(k)$, $d(k)$ are functions of $d_{CH,BS}(k)$, they will converge as well. $\square$

Remark 3.1. We select the sensor nearest the optimal position derived from our algorithm as the CH node. Although this cannot guarantee to achieve optimal solution, we use extensive simulations to demonstrate its near-optimality in Section 3.5.

Remark 3.2. We derived our weight expression based on the idea that $P_{CH} = P_{NCH_f}$. We prove the convergence of our algorithm. It can be shown that in most of deployment cases, after we run this algorithm, we could get $P_{CH} = P_{NCH_f}$. Furthermore, our algorithm can also be used to deal with the case when NCHs are deployed quite close to BS, i.e., even if we put CH at BS, we still have $P_{CH} > P_{NCH_f}$. The convergence of our algorithm is not affected according to our proof.

3.2.2 Cluster Formation Algorithm ($N_c > 1$, $\alpha = \frac{1}{N_s}$)

In a large-scale network, it may be beneficial to partition it into N_c clusters ($N_c > 1$), where N_c CH nodes are able to fuse information from NCHs and forward the information to the BS. Such an N_c -cluster network model is depicted in Figure 3.2. Our problem is defined as follows: Given N_c , how do we partition a network with N_s nodes into N_c clusters and determine the respective CH nodes, denoted by $CH_1, CH_2, \dots, CH_{N_c}$, to maximize the overall network lifetime, i.e., the time till the first node dies?

We begin by considering the case where N_c CH nodes have been given, and the task is to join NCHs into these N_c clusters with maximized network lifetime. Denote the CH nodes as CH_j , $1 \leq j \leq N_c$. The power of CH_j is given as follows:

$$P_{CH_j} = E_{elec}r(N_{s,j} - 1) + E_{DA}rN_{s,j} + E_{elec}r + E_{amp}d_{CH_j,BS}^2r \quad (3.4)$$

Figure 3.2: Our proposed multiple cluster WSN model.

The main idea is to join one NCH into a certain cluster in each iteration until all the sensors have joined the clusters. In each iteration, let us denote by NCH_j^* the sensor that is closest to CH_j among all the sensors which have not joined any cluster yet and $P_{NCH_j^*}$ the corresponding power consumption rate if the sensor does join cluster j . Obviously, if any sensor is to join cluster j in this iteration, it should be NCH_j^* since it leads to the smallest increase in $NCH_{f,j}$. We repeat this process for all CHs, determine $j^* = \arg \min_j P_j$, where $P_j = \max(P_{NCH_j^*}, P_{CH_j})$ where P_{CH_j} is calculated based on the assumption that one more sensor is joining cluster j , and assign $NCH_{j^*}^*$ to CH_{j^*} . This process is repeated until all NCHs join the clusters.

We now describe the multi-cluster formation algorithm. Randomly select N_c nodes as initial CHs and use the algorithm above to join all the other sensors into these clusters. Denote the network lifetime as $L(0)$. We then apply our cluster head selection algorithm as described in Section 3.2.1 to select new CHs in each cluster independently; repeat the cluster formation process described above, and denote the network lifetime as $L(1)$. Repeat the cluster formation and cluster head selection algorithms iteratively until $|L(k) - L(k-1)| < \epsilon$. The proposed algorithm for an N_c -cluster network is shown in Algorithm 3.2. We illustrate the algorithm for a simple example of $N_c = 2$ in Figures 3.3-3.5.

It can be easily proven that Algorithm 3.2 converges. Specifically, whenever step 2 is executed, the network lifetime either remains unchanged or is increased (when a better candidate is found). The lifetime may be further improved in step 3 when every NCH joins the selected CH to maximize lifetime. In short, the overall network lifetime increases progressively until a local optima is found.

3.3.a

3.3.b

Figure 3.3: Step-by-step illustration of the cluster formation algorithm (a) CH_1 and CH_2 are randomly selected from existing sensors to serve as CHs, NCH_1^* and NCH_2^* are closest NCHs from CH_1 and CH_2 respectively. If NCH_1^* joins CH_1 , we denote $P_1 = \max(P_{CH_1}, P_{NCH_1^*})$. If NCH_2^* joins CH_2 , we denote $P_2 = \max(P_{CH_2}, P_{NCH_2^*})$. The dotted line is used to show the closest NCH for each CH. (b) If $P_1 < P_2$, NCH_1^* in Figure 3.3.a joins CH_1 . We use a solid line to show the connection. Then we find new closest NCHs from both CHs, still denoted as NCH_1^* and NCH_2^* , which are connected to their respective CHs through dotted lines.

3.4.a

3.4.b

Figure 3.4: Step-by-step illustration of the cluster formation algorithm (a) Assume $P_1 > P_2$ in Figure 3.3.a) and a node joins CH_2 , shown as the solid line. We find new closest NCHs from both CHs, which are connected to their respective CHs via dotted lines. (b) We repeat the above process for each NCH one by one until all NCHs join the corresponding CHs. We denote the whole network lifetime as L_0 .

Remark 3.3. Starting from a set of randomly selected CHs, Algorithm 3.2 can only ensure achieving a local optimal solution. Running the algorithm from a large enough number of different initial sets of CHs helps improve the likelihood that global optimality or sub-optimality

Figure 3.5: Cluster head selection + cluster formation (a) We run Algorithm 3.1 for each cluster independently to find the new CHs, denoted as CH_1 and CH_2 . (b) Cluster formation by using steps shown in Figure 3.3 and Figure 3.4. Denote the network lifetime as L_1

is achieved. We demonstrate in Section 3.5 that our algorithm's performance is close to optima in most cases.

3.2.3 Dynamic Clustering Algorithm ($N_c \geq 1$, $\alpha = \frac{1}{N_s}$)

As discussed in Section 2.1.2, since there are no existing results with the same network model and objective function as ours, we extend our algorithm to dynamic clustering for comparison. Specifically, we assume, as in most existing results, that re-clustering is carried out at the beginning of each round of sensing. Note that, for simplification, we neglect any possible energy consumption involved in cluster formation and re-formation. To allow the proposed algorithm to handle dynamic clustering, the minimum change needed is to handle the case where different sensors may have different initial energies. Specifically, the changes we make are as follows:

- Recall that in the cluster head selection algorithm presented in Section 3.2.1, we assign weight 1 to sensors and w_0 to BS according to (3.3) and iteratively update w_0 in the k^{th} iteration (denoted as $w(k)$). This is based on the assumption that all the sensors are equipped with same energy. Now we consider the case where every sensor is equipped with energy E_i , which may be different for different i . We need to derive the new weight expression for sensors such that at the optimal location of CH, $\min(\frac{E_{CH}}{P_{CH}}, \frac{E_{NCH_f}}{P_{NCH_f}})$ is maximized. It is easy to see that at the optimal CH position, we have $\frac{E_{CH}}{P_{CH}} = \frac{E_{NCH_f}}{P_{NCH_f}}$. By

Algorithm 3.2 Cluster Formation Algorithm, $N_c > 1$

- **Initialization**

Input the set of sensor locations $A = (x_1, y_1), (x_2, y_2) \dots (x_{N_s}, y_{N_s})$. Let $(x_0, y_0) = (0, 0)$ be the coordinates for the BS. Randomly select N_c CHs from the existing N_s sensors, labeled as $CH_1, CH_2, \dots, CH_{N_c}$. Set iteration number $k = 0$.

- **Step 1**

For each CH_j , determine P_j if the closest unassigned NCH, denoted by NCH_{j^*} , is added to it. Find $j^* = \arg \min_j P_j$ and add NCH_{j^*} to cluster j^* . Repeat until all NCH nodes are assigned to clusters. Denote the network lifetime as $L(k)$.

- **Step 2**

Use Algorithm 3.1 to find the new CH position independently for each cluster, and increment k .

- **Step 3**

Repeat the process in Step 1 until all the nodes join the clusters. We record the lifetime of the network as $L(k)$.

- **Step 4**

Compare $L(k)$ with $L(k - 1)$: if $|L(k) - L(k - 1)| > \epsilon$, then goto Step 2; otherwise, terminate the algorithm.

adopting the same approach as shown in Section 3.2.1, we have:

$$w_0 = \sqrt{\frac{E_{elec}(N_s - 1) + E_{DA}N_s}{E_{amp}d_{CH,BS}^2} + 1} \quad (3.5)$$

$$w_i = \sqrt{\frac{E_{CH}}{E_i}}. \quad (3.6)$$

By doing so, when $w_0 d_{CH,BS} = w_{NCH_f} d_{CH,NCH_f}$, we have $\frac{E_{CH}}{P_{CH}} = \frac{E_{NCH_f}}{P_{NCH_f}}$. Once again, only w_0 needs to be updated in each iteration. Hereafter we always use $w(k)$ to denote the value of w_0 in the k^{th} iteration.

As E_{CH} is not known in advance, a convenient option is to let $E_{CH} = \frac{\sum_{i=1}^{N_s} E_i}{N_s}$ though setting E_{CH} at any other positive values would hardly affect the algorithm efficiency. After the CH selection algorithm finds the optimal CH position, among sensors that

have energy larger than the average energy of sensors in the cluster, the one closest to the optimal position is chosen as CH. In this way, if the sensor closest to the optimal position has low residual energy, its energy can be conserved by not serving as CH. The convergence of the algorithm can also be proven using the theorems shown in Section 3.2.1.

- Recall that for the cluster formation algorithm shown in Section 3.2.2, we determine $j^* = \arg \min_j P_j$ based on the assumption that the initial energy is the same in different sensors. When this is not true, both E_i and P_i have to be considered in assigning sensors into clusters to optimize lifetime. Specifically, in each iteration, we still add one sensor into one of the clusters: given N_c CHs, for the closest NCHs from these CHs, we determine $j^* = \arg \max_j L_j$, where $L_j = \min(E_{NCH_j^*} - P_{NCH_j^*}T, E_{CH_j} - P_{CH_j}T)$. T is a constant, which is the duration of each time slot in the TDMA schedule. Then we add $NCH_{j^*}^*$ into CH_{j^*} , and repeat this process until all NCHs join the respective CHs. By adopting an approach similar to that in Section 3.2.2, the convergence of the modified cluster formation algorithm can be proven.

Note that the extended algorithm is not optimized for dynamic clustering. Instead, it is the algorithm for static clustering with some minimum changes. Nevertheless, as we will see later in Section 3.5, the extended algorithm outperforms all the existing ones.

3.3 Algorithms for Maximizing Lifetime of WSNs with EH Sensors ($\alpha = \frac{1}{N_s}$)

In this section, we assume the availability of EH nodes that can harvest and store energy from the environment, and study how they can be exploited in a clustered WSN to maximize network lifetime. In general, each EH node can operate in three different ways: (i) serves as relay for CH, i.e., receives the data from CH and forwards it to BS; (ii) serves as CH, i.e., receives, aggregates information from all NCHs and forwards to BS whenever it has available energy; or (iii) serves as a relay between NCHs and CH node. In this thesis, we consider the case where N_c EH nodes are available as depicted in Figure 3.2, and focus on Case (i), where each EH node, EH_j , serves as a dedicated relay node for CH_j , leaving the remaining cases to future

work. Our objective is to study the joint placement of CH and EH nodes to maximize the lifetime of WSNs.

The main idea of the proposed algorithm is that, for each cluster j , we find out the *relationship* between the best location of CH_j and the best location of EH_j . Then by treating EH_j as if it were the BS and assigning a proper weight to it, we find the best location of CH_j and correspondingly, the best location of EH_j as well.

We begin with a single cluster WSN as shown in Figure 3.1, and let E_s and $\hat{t}$ denote the initial energy and the lifetime of the CH node respectively. Over the duration of $\hat{t}$, the amount of energy that the EH node (assumed to have zero initial energy) can harvest is given by $P_{EH,h}\hat{t}$. Since it consumes energy at a rate of $P_{EH,c}$, it can remain active and relay data for the CH for a duration of t_{EH} , where

$$t_{EH} = \min\left(\frac{P_{EH,h}\hat{t}}{P_{EH,c}}, \frac{E_s}{P_{CH,on}}\right),$$

and the latter term is the lifetime of the CH node when it transmits via the EH node.

During this period, the energy depleted in the CH node is given by $P_{CH,on}t_{EH}$. Subsequently, the EH node is inactive, and the CH node transmits directly to BS for a duration of $\frac{E_s - P_{CH,on}t_{EH}}{P_{CH,off}}$ until its energy is depleted. Accordingly, we obtain the expression for $\hat{t}$ as:

$$\hat{t} = \begin{cases} \frac{P_{EH,h}\hat{t}}{P_{EH,c}} + \frac{E_s P_{EH,c} - P_{CH,on} P_{EH,h} \hat{t}}{P_{CH,off} P_{EH,c}}, & \hat{t} < \frac{E_s P_{EH,c}}{P_{EH,h} P_{CH,on}}; \\ \frac{E_s}{P_{CH,on}}, & \text{otherwise.} \end{cases} \quad (3.7)$$

- **Case 1:** $\hat{t} < \frac{E_s P_{EH,c}}{P_{EH,h} P_{CH,on}}$

According to (3.7), we have

$$\hat{t} = \frac{E_s P_{EH,c}}{P_{EH,c} P_{CH,off} + P_{CH,on} P_{EH,h} - P_{EH,h} P_{CH,off}} \quad (3.8)$$

We express $P_{CH,off}$, $P_{CH,on}$ and $P_{EH,c}$ in terms of $d_{CH,BS}$ and $d_{CH,EH}$ in a similar way as that in (3.1), and they are given as follows:

$$P_{EH,c} = E_{elec}r + E_{amp}d_{EH,BS}^2 \quad (3.9)$$

$$P_{CH,on} = E_{elec}rN_s + E_{DA}rN_s + E_{amp}d_{CH,EH}^2 \quad (3.10)$$

$$P_{CH,off} = E_{elec}rN_s + E_{DA}rN_s + E_{amp}d_{CH,BS}^2 \quad (3.11)$$

According to the rule of triangularity, if EH is not located on the line connecting CH and BS, we have $d_{CH,EH} \geq d_{CH,BS} - d_{EH,BS}$. Accordingly, (3.10) becomes:

$$P_{CH,on} \geq E_{elec}rN_s + E_{DA}rN_s + E_{amp}(d_{CH,BS} - d_{EH,BS})^2r \quad (3.12)$$

According to (3.12), to keep $P_{CH,on}$ low, the EH node should be placed on the line connecting CH and BS. Substituting (3.9), (3.11) and (3.12) into (3.8), we obtain

$$\hat{t} = \frac{E_s}{E_{elec}rN_s + E_{DA}rN_s + E_{amp}d_{CH,BS}^2r - f(d_{EH,BS})} \quad (3.13)$$

where $f(d_{EH,BS}) = \frac{(2d_{CH,BS} - d_{EH,BS})P_{EH,h}}{d_{EH,BS} + \frac{E_{elec}}{E_{amp}d_{EH,BS}}}$.

In order to maximize $\hat{t}$, according to (3.13), $f(d_{EH,BS})$ should be maximized. Since $\hat{t} < \frac{E_s P_{EH,c}}{P_{EH,h} P_{CH,on}}$, substituting (3.8), it can be easily deduced that $P_{EH,h} < P_{EH,c}$. Applying (3.9), we have

$$P_{EH,h} \leq E_{elec}r + E_{amp}r d_{EH,BS}^2$$

$\Rightarrow$

$$d_{EH,BS} > \sqrt{\frac{P_{EH,h} - E_{elec}r}{E_{amp}r}}$$

If we denote $d_{EH,BS}^* = \sqrt{\frac{P_{EH,h} - E_{elec}r}{E_{amp}r}}$, and assume that $P_{EH,h} \gg E_{elec}r$, then $d_{EH,BS}^* > \sqrt{\frac{E_{elec}}{E_{amp}}}$. Considering $f(d_{EH,BS})$, we note that when $d_{EH,BS} = d_{EH,BS}^*$, $\frac{2d_{CH,BS}}{d_{EH,BS} + \frac{E_{elec}}{E_{amp}d_{EH,BS}}}$ is maximized while $\frac{d_{EH,BS}}{d_{EH,BS} + \frac{E_{elec}}{E_{amp}d_{EH,BS}}}$ is minimized. Thus, $f(d_{EH,BS})$, and hence $\hat{t}$, is maximized when $d_{EH,BS} = d_{EH,BS}^*$.

- **Case 2:** $\hat{t} \geq \frac{E_s P_{EH,c}}{P_{EH,h} P_{CH,on}}$

According to (3.7), we have:

$$\hat{t} = \frac{E_s}{P_{CH,on}} \quad (3.14)$$

Taking (3.12) into (3.14), we have

$$\hat{t} = \frac{E_s}{E_{elec}rN_s + E_{DA}rN_s + E_{amp}(d_{CH,BS} - d_{EH,BS})^2r} \quad (3.15)$$

Since $\hat{t} \geq \frac{E_s P_{EH,c}}{P_{EH,h} P_{CH,on}}$, applying (3.14) and (3.9), we have $P_{EH,h} \geq P_{EH,c}$ and $d_{EH,BS} \leq d_{EH,BS}^*$. According to (3.15), $\hat{t}$ can be maximized if $d_{EH,BS} = d_{EH,BS}^*$.

In summary, $\hat{t}$ is maximized when the EH node is placed on the line joining the BS and CH nodes such that $d_{EH,BS} = d_{EH,BS}^*$. At this position, we have that for the EH sensor, the energy consumption rate equals its energy harvesting rate. The corresponding maximum lifetime can be expressed as:

$$\hat{t}_{max} = \frac{E_s}{E_{elec}rN_s + E_{DA}rN_s + E_{amp}r(d_{CH,BS} - d_{EH,BS}^*)^2}$$

Next, to determine the CH position, we use the same approach as shown in Section 3.2.1 by letting $P_{CH} = P_{NCH_f}$, where

$$P_{CH} = P_{CH,off} \frac{\hat{t} - t_{EH}}{\hat{t}} + P_{CH,on} \frac{t_{EH}}{\hat{t}},$$

and P_{NCH_f} is given in (3.2). We apply Algorithm 3.1 to determine the optimal CH position, (x^*, y^*) , with the following expression for the weight for EH:

$$w(k) = \sqrt{\frac{E_{elec}(N_s - 1) + E_{DA}N_s}{(E_{amp}d_{CH,BS}(k-1)^2)} + \frac{(d_{CH,BS}(k-1) - d_{EH,BS}^*)^2}{d_{CH,BS}(k-1)^2}}, \quad (3.16)$$

and the corresponding location of the EH node, (x_{EH}, y_{EH}) , is then given by:

$$(x_{EH}, y_{EH}) = (d_{EH,BS}^* \frac{x^*}{\sqrt{x^{*2} + y^{*2}}}, d_{EH,BS}^* \frac{y^*}{\sqrt{x^{*2} + y^{*2}}}) \quad (3.17)$$

The algorithm for a single cluster with EH node is presented in Algorithm 3.3. As those in Section 3.2.1, we can prove that $d_{CH,NCH_f}(k)$, $d_{CH,BS}(k)$ and $w(k)$ converge and the efficiency of convergence is also verified through simulations.

Remark 3.4. We note that the EH node is placed at (x_{EH}, y_{EH}) , which is computed based on (x^*, y^*) . For simplicity, we let it remain in this position when the node nearest to (x^*, y^*) is chosen as the CH. The optimal location of the EH sensor, if needed, can be computed to be on the straight line between BS and CH with its energy consumption rate equaling its energy harvesting rate, which is typically quite close to (x_{EH}, y_{EH}) in a cluster with a reasonably high density of sensors.

The extension to the multi-cluster case is obtained by introducing the cluster index, j , to the corresponding expressions for $d_{EH,BS}^*$ and $w(k)$ and applying Algorithm 3.2 in Section 3.2.2. The corresponding procedure is given in Algorithm 3.4.

Algorithm 3.3 Single Cluster Algorithm with EH node

- **Initialization**

Input set of sensor locations $A = (x_1, y_1), (x_2, y_2) \dots (x_{N_s}, y_{N_s})$. Let $(x_0, y_0) = (0, 0)$ which represents the coordinates for the BS, and set $w(1) = 1$;

- **Step 1**

Run Algorithm 3.1 using the weight expression in (3.16) instead of (3.3).

- **Step 2**

Record the CH position from the algorithm as (x^*, y^*) (not from existing sensors). The EH node position (x_{EH}, y_{EH}) is then given by (3.17).

- **Step 3**

We select the sensor that is closest to (x^*, y^*) as CH and place the EH node at position (x_{EH}, y_{EH}) .

Similar to that for Algorithm 3.2 in Section 3.2.2, it can be easily proven that Algorithm 3.4 converges.

For dynamic clustering schemes, the algorithms can be extended similarly as that in Section 3.2.3 to handle the case where different sensors have different initial energies. Specifically, for the cluster head selection algorithm, we assign a weight $w_i = \sqrt{\frac{E_{CH}}{E_i}}$ to each sensor. When the optimal position for CH has been found, we select the closest sensor from this position with energy larger than the average energy in the cluster to serve as CH. For the cluster formation algorithm, we adopt the same procedure as that in Section 3.2.3.

3.4 Algorithms for Maximizing Lifetime of WSNs with EH Sensors ($\frac{1}{N_s} < \alpha < 1$)

Some networks may still function well even when a portion of nodes have died [2]. Therefore, we also consider the case when the network lifetime is the time duration until a portion of nodes (defined as $\alpha\%$) die. In this section, we generalize our previous algorithms to the case where $\frac{1}{N_s} < \alpha < 1$. Our problem can be stated as follows: for an N_c -cluster WSN with EH nodes, how do we form the clusters and place the CH and EH nodes to maximize the overall network lifetime, i.e., the time until αN_s nodes die?

Algorithm 3.4 Multiple Cluster Algorithm with EH nodes

- **Initialization**

Input set of sensor locations $A = (x_1, y_1), (x_2, y_2) \dots (x_{N_s}, y_{N_s})$. Let $(x_0, y_0) = (0, 0)$ which represents the coordinates for the BS.

- **Step 1**

Run Algorithm 3.2 for the network using weight expression as shown in (3.16) for EH_j .

- **Step 2**

Record the position for CH_j from the algorithm as $(x_{CH_j}^*, y_{CH_j}^*)$ where $1 \leq j \leq N_c$ (not from existing sensors). Position for EH_j can be calculated as $(d_{EH_j, BS}^* \frac{x_{CH_j}^*}{\sqrt{x_{CH_j}^{*2} + y_{CH_j}^{*2}}},$

$$d_{EH_j, BS}^* \frac{y_{CH_j}^*}{\sqrt{x_{CH_j}^{*2} + y_{CH_j}^{*2}}}).$$

- **Step 3**

Select the sensor in cluster j that is closest from $(x_{CH_j}^*, y_{CH_j}^*)$ to serve as CH for cluster j , other sensors remain as NCHs. N_c EH nodes are deployed at the positions shown in step 2.

The intuition behind the algorithm is similar to that for the cluster formation algorithm shown in Section 3.2.2. We begin by assuming that N_c CH nodes have been assigned and the task is to form N_c clusters (i.e., assign NCHs to clusters) with the maximized lifetime. The lifetime is defined as the time until $N_s \alpha$ nodes die. In other words, the lifetime is the time duration that $N_s(1 - \alpha)$ nodes can function. As in Section 3.2.2, we assign one NCH to a cluster in every iteration. The only difference is that the process is repeated only until $N_s(1 - \alpha)$ closest nodes are covered. By doing this, we maximize the time that $N_s(1 - \alpha)$ nodes can function, thus maximizing the network lifetime.

We now describe the multi-cluster formation algorithm for the general lifetime definition. We randomly select N_c nodes as initial CHs and use the algorithm above to join $N_s(1 - \alpha)$ closest nodes into these clusters. Denote the network lifetime as $L(0)$. We then apply our cluster head selection algorithm as described in Section 3.2.1 to select new CHs in each cluster independently; repeat the cluster formation process described above, and denote the network lifetime as $L(1)$. Repeat the cluster formation and cluster head selection algorithms iteratively until $|L(k) - L(k - 1)| < \epsilon$.

Algorithm 3.5 Multiple Cluster Algorithm with EH node for general α

- **Initialization**

Input set of sensor locations $A = (x_1, y_1), (x_2, y_2) \dots (x_{N_s}, y_{N_s})$. Let $(x_0, y_0) = (0, 0)$ be the coordinates for the BS. Input $0 < \alpha < 1$. Randomly select N_c CHs from the existing N_s sensors, labeled as $CH_1, CH_2, \dots, CH_{N_c}$. Set iteration number $k = 0$.

- **Step 1**

For each CH_j , determine P_j if the closest unassigned NCH, denoted by NCH_j^* , is added to it. Find $j^* = \arg \min_j P_j$ and add $NCH_{j^*}^*$ to cluster j^* . Repeat until $N_s(1 - \alpha)$ nodes are assigned to clusters. Denote the network lifetime as $L(k)$.

- **Step 2**

Use Algorithm 3.3 to find the new CH position for each cluster, and increment k .

- **Step 3**

Repeat the process in Step 1 until $N_s(1 - \alpha)$ nodes are assigned to the clusters. We record the lifetime of the network as $L(k)$.

- **Step 4**

Compare $L(k)$ with $L(k - 1)$: if $|L(k) - L(k - 1)| > \epsilon$ then goto Step 2; otherwise, we terminate the algorithm. The j^{th} EH position is then given by $(d_{EH_j, BS}^* \frac{x_{CH_j}^*}{\sqrt{x_{CH_j}^{*2} + y_{CH_j}^{*2}}},$

$$d_{EH_j, BS}^* \frac{y_{CH_j}^*}{\sqrt{x_{CH_j}^{*2} + y_{CH_j}^{*2}}}).$$

Our proposed algorithm is shown in Algorithm 3.5. Again, it can be easily proven that Algorithm 3.5 converges.

3.5 Simulation Results

We demonstrate the performance of our proposed algorithms through simulations using Qualnet for a 2-D network with a base station deployed at (0,0) and $N_s = \{50, 100, 150, 200\}$ nodes randomly distributed over a square region, each initially equipped with 0.5 J of energy and has a data transmission rate of 2000 bits per packet. We consider three square regions of network deployment, where the coordinates of the vertices are as follows: **Case I** (200, 200), (200, 300), (300, 200) and (300, 300); **Case II** (100, 100), (100, 200), (200, 100), (200, 200); **Case**

Table 3.2: Comparison of network lifetime for various cases ($N_s = 100$, $N_c = 1$)

	Case I	Case II	Case III
Brute force	17.99	32.41	35.05
Our algorithm	17.99	32.41	33.88

III (0, 0), (0, 300), (300, 0), (300, 300). Although Cases I and II have the same area, nodes are located closer to the BS in Case II than Case I. Cases I and III differ in terms of node density. By using these three cases, we can determine the influences of proximity to the BS and node density on the performance of the proposed algorithms.

The performance metric used in this thesis is the number of rounds the network could operate, where a round is a TDMA frame composed of N_s times slots, one for each sensor node. After cluster formation and placement of EH nodes (where applicable), we repeat the TDMA frame until $\alpha\%$ of nodes dies. In Section 3.5.1, we verify that our clustering algorithms (with complexity $O(N_s)$) can achieve near-optimal network lifetime for the case of $\alpha = \frac{1}{N_s}$. We do so by bench-marking against a brute force method, where the optimal CH position is obtained through exhaustive search (with complexity $O(N_s^2)$). We also characterize the multi-cluster algorithms in terms of the optimal number of clusters for a given network. In Section 3.5.2, we analyze the effects of EH nodes, specifically the energy harvesting rates, on the network lifetime. In Section 3.5.3, we study the performance of our proposed clustering algorithms for general α .

3.5.1 Performance of Clustering Algorithms without EH Nodes

- $N_c=1$

The result shows that brute force method and our algorithm can achieve the same result with the effects of node density. The effects of proximity to the BS on the algorithm performance is shown in Tables 3.2 respectively. The results, obtained by averaging over 150 runs, show that our algorithm can achieve performance very close (up to 3.34 %) to the optimal case (brute force).

We then observe the rate of convergence for our algorithm, which is shown in Figure 3.6. The results clearly show fast convergence (usually within 10 iterations). In this thesis, we

Figure 3.6: Comparison of convergence efficiency of our algorithm for all cases ($N_s = 100$).

Table 3.3: Comparison between our method and brute force method ($N_c = \{2,3\}$, $N_s = \{100,150\}$)

	(N_s, N_c)			
	(100,2)	(150,2)	(100,3)	(150,3)
Brute force	49.28	39.72	60.42	51.20
Our algorithm	49.28	39.72	58.68	50.83

only present results for Cases I and II with $N_s = \{100,150\}$. Note that our conclusions hold for all the other cases.

- $N_c > 1$, Cases I, II, $N_s = \{100,150\}$

We compare the network lifetime for $N_c = \{2,3\}$ in Table 3.3. Once again, our proposed algorithm can achieve performance close to that of the brute force method with lower time complexity – this gain becomes significant as N_c increases.

We then compare the performance of the extended algorithm with the existing dynamic clustering algorithms. For fair comparison with LEACH, MOECS, AEEC and pLEACH, we adopt the same optimal number of clusters shown in LEACH [13]. We run simulations with 25 different network configurations for both Case I and Case II from 100 to 150 nodes in steps of 10. The result is plotted in Figures 3.7(a) and 3.7(b) for both Case I and Case II.

We find that our algorithm achieves the highest lifetime compared with the other approaches. Specifically, for Case II with 100 sensors, our algorithm improves the lifetime by 33.03% compared with LEACH and 3.31% compared with pLEACH. The improvement is less significant compared with pLEACH because pLEACH divides the area into several sectors in order to minimize the transmission distance from sensors to CHs as well as to balance the cluster size. Our work uses a simpler way without dividing the area into several sectors, and yet can still outperform pLEACH. For Case I with 100 sensors, our algorithm improves the lifetime by 56.84% compared with LEACH and 18.4% compared with pLEACH. For Case II, we find that with the increase in the number of sensors, the lifetime does not always increase. This is because the optimal number of clusters varies with the number of sensors [13], thus affecting the network lifetime.

Next, we investigate the performance of our proposed algorithm for $N_c = \{1, 2, \dots, 10\}$ as shown in Figure 3.8. For Case II, network lifetime is maximized with 5 and 6 clusters when $N_s = 100$ and 150 respectively; for Case I, network lifetime is maximized where there are 4 and 5 clusters respectively. The differences in the optimal numbers of clusters for these two cases are due to the fact that sensors in Case I are farther from BS than those in Case II. Consequently, the optimal number of clusters becomes lower to have fewer CHs which have to communicate with BS over long distances. In other words, when CHs are far away from BS, the distance between BS and CH has stronger impact on network lifetime than the number of sensors in each cluster.

It is well-known that the overall energy consumption is a convex function of the number of clusters (e.g., [13]). It is interesting, though not a big surprise to see that the network lifetime appears to be a concave function of the number of clusters: when N_c is very small (say, $N_c = 1$ in the extreme case), the CH needs to receive data from all NCHs, and therefore network lifetime is determined by this CH. When N_c is very large (e.g., $N_c = N_s$ in the extreme case), all NCH nodes will transmit directly to the BS, and hence the network lifetime is determined by the NCH which is farthest from the BS. Therefore, as N_c increases, the energy consumption rate of the CH node is reduced (i.e., network lifetime increases) initially due to the reduction in cluster size; however, with further increase in N_c , the energy consumption rates in some CH nodes increase because the distance between these CHs and BS increases, which reduces the network lifetime.

3.7.a

3.7.b

Figure 3.7: Comparison between our proposed algorithm with selected algorithms for (a) Case I; (b) Case II.

3.5.2 Performance of Joint EH placement and Clustering Algorithm

Next, we study the performance of our proposed joint EH placement and clustering algorithm, where each CH node is served by a unique EH node, for the range of EH rates between $0.03W$ and $0.09W$, in steps of $0.02W$. The energy harvesting rate is assumed according to the previous literature [4, 120], typically tens of milliwatts.

3.8.a

3.8.b

Figure 3.8: Network lifetime vs number of clusters for our proposed algorithm ($N_c = \{1, 2, \dots, 10\}$).

- $N_c=1$

We quantify the impact of the EH node on the network lifetime in Figure 3.9. For a single cluster network, introducing an EH node with a charging rate of 30 mW results in a gain of 8-13 % in network lifetime for different cases. Certainly the improvements depend on many factors including energy harvesting rate, network scale, and more. As

expected, for a given node density, the network lifetime decreases when the network is farther from the BS; for a given deployment, the network lifetime is reduced when the node density increases.

Figure 3.9: Effects of using EH node on the network lifetime ($N_c = 1$, $P_{EH,h} = 0.03W$).

- $N_c > 1$

3.5.2.1 Characterization of Multiple Cluster Algorithm with EH Sensors

We investigate the performance of the proposed algorithm for Case II with $N_c = \{1, 2, \dots, 10\}$, as shown in Figure 3.10. Similar to that in Section 3.5.1, the network lifetime is maximized with 5 and 6 clusters when $N_s = 100$ and 150 respectively. The optimal number of clusters for each case remains invariant for energy harvesting rate varying from 30 mW to 90 mW. This can be explained as follows: in our simulations, we adopt relatively high data transmission rate of $2.5 \times 10^5 b/s$ and typical harvesting rates of 30-90 mW. As a result, the harvesting rate of EH is much lower than the energy consumption rate of CH. Hence the optimal location of EH is typically much closer to BS than the CH. For example, for Case II, $N_c = 5$ and $N_s = 100$, the optimal location of EH is 26.46 m away from BS while the closest CH from BS is 151.43 m away. Since the optimal number of clusters is largely determined by distance between CH and BS, the existence of EH nodes therefore does not change the optimal number of clusters.

Note that for the case where CHs have a lower energy consumption rate or EH has a much higher energy harvesting rate, we can expect the optimal number of clusters to increase. For the extreme case where EH nodes have an infinite energy harvesting rate, it is optimal to co-locate the EH and CH nodes and the optimal number of clusters equals the number of sensors. This unrealistic case nevertheless illustrates the trend of the optimal number of clusters when the EH harvesting rate goes up.

Figure 3.10: Network lifetime vs number of clusters for our proposed algorithm ($N_c = \{1, 2, \dots, 10\}$, $P_{EH,h} = 0.03W$)).

For Case II, with N_c being set to the optimal number of clusters for $N_s = 100$ and 150 respectively, we quantify the gain in network lifetime achievable at various energy harvesting rates in Figure 3.11. The result is average value over simulations of 30 different network topologies with 100 simulations over each network topology. We observe that the network lifetime increases with energy harvesting rate. Specifically, in Case II, with the existence of EH nodes with a harvesting rate of 30mW, the lifetime of the network increases from 70.05 to 87.58, leading to a 25.02 % increase. Note that this increase is achieved when CHs are transmitting at a relatively high bit rate. If CHs are transmitting at a lower bit rate, the increase will be more significant.

For a given number of clusters, a continuous increase in the EH sensor harvesting rate leads to a less and less significant further improvement in network lifetime when EH moves towards CH. The lifetime will finally reach a constant value where further increasing harvesting rate does not help. By then the cluster lifetime is decided by the

3.11.a

3.11.b

Figure 3.11: Effects of energy harvesting rate on network lifetime for different cases with optimal number of clusters for (a) Case I; (b) Case II.

distance between CH and the farthest NCH, which cannot be arbitrarily shortened by the existence of the EH node.

We finally evaluate the effects of energy harvesting rate on the network lifetime using the dynamic clustering algorithm. The results are shown in Figures 3.12(a) and 3.12(b) for both Case I and Case II when the number of clusters is still set to be the optimal number

3.12.a

3.12.b

Figure 3.12: Effects of energy harvesting rate on network lifetime using dynamic clustering algorithm for (a) Case I; (b) Case II.

of clusters of LEACH. For both Case I and Case II, we find that the lifetime is improved significantly when the harvesting rate is increased from 0 to 0.03 W. When the harvesting rate continues to increase, once again we find that the network lifetime is improved less significantly, for the same reason given earlier.

3.5.3 Algorithm Performance for Different Lifetime Definitions

Lastly, we illustrate the performance of our proposed joint EH placement and clustering algorithms for different values of α ranging from $\{0.1, 0.2, \dots, 0.5\}$ for Case II, with $N_c = 3$. We compare the network lifetime with the brute force (optimal) method for clustering, and plot the results in Figure 3.13, with the first point on the x -axis corresponding to $\alpha = \frac{1}{N_s}$.

As expected, the network lifetime increases as α (i.e., the permissible number of dead nodes) increases. We also observe that our algorithm is able to achieve performance sufficiently close to the brute force method (within 10%).

Figure 3.13: Comparison between our algorithm with brute force method at various α (Case II, $N_c = 3$, $N_s = \{100, 150\}$)

3.6 Conclusion

This chapter considered clustered wireless sensor networks (WSN) where CHs either aggregate and forward data directly to BS, or via dedicated relay nodes with energy harvesting (EH) capabilities. We proposed efficient (polynomial-time) EH node placement and clustering algorithms to maximize network lifetime, where the network lifetime is the duration until a certain $\alpha\%$ of the nodes run out of energy. Through theoretical analysis and extensive simulations, we validated the near optimality of the proposed algorithms and demonstrated how much EH sensors can help prolong network lifetime in different scenarios. In addition, we showed the

existence of an optimal numbers of clusters for a given network configuration (defined by the node density and proximity of the network to the BS), which may not be significantly changed by the existence of EH sensors with typical energy harvesting rates.

As mentioned in Section 1.1.2, centralized algorithms could provide benchmarks for distributed algorithms where sensors have limited computational capabilities. In Chapter 4, we shall shift the focus to designing distributed clustering algorithms for WSNs.

Chapter 4

Distributed Clustering Algorithms for Maximizing Lifetime

This chapter presents distributed clustering algorithms for multi-clustered networks, with and without EH relay sensors respectively, with the objective of maximizing the network lifetime. Section 4.1 presents a distributed lifetime-optimal relay scheduling algorithm for multi-clustered WSN with a single battery powered relay node. In Section 4.2, we extend the algorithms to include multiple relay nodes with energy harvesting capabilities. In section 4.3, extensive simulation results and discussions are presented for verifying the performance of the proposed algorithms. Section 4.4 concludes this chapter.

4.1 Lifetime-Optimal Single Battery Powered Relay Node Scheduling for Multi-clustered WSN

In this section, we design distributed clustering algorithm with single battery powered relay node. Specifically, we propose the system model in section 4.1.1. Then we propose the distributed algorithm in section 4.1.2.

4.1.1 System Model

In this section, we consider a homogeneous and fixed clustered WSN deployment in a 2-D region that comprises a mains-powered base station (BS) located at (0,0), N_c CH nodes and $N_s - N_c$ NCH nodes deployed randomly, and a relay node deployed between the sensor nodes and the BS. We assume that the CH nodes have been selected, and clusters have been formed.

Our network model is depicted in Figure 4.1. We also assume that the network is sufficiently far away from the BS such that the network lifetime is given by the time till the first CH node runs out of energy.

Table 4.1: Notations used throughout section 4.1

Notation	Description	Value
N_s	Number of sensors in the network	N.A.
N_c	Number of clusters in the network	10
i	CH index	$\{1, 2, \dots, 10\}$
j	Index of iteration	N.A.
E_{CH_i}	residual energy of CH_i	$(0, 0.5J)$
E_s	Energy stored in battery for each sensor	0.5J
$P_{CH_i, off}$	Energy consumption rate for CH_i when relay node is not working	N.A.
$P_{CH_i, on}$	Energy consumption rate for CH_i when relay node is working	N.A.
ΔE	Overhead energy consumption for relay node to select CH	N.A.
$L_{CH_i}^{direct}$	lifetime for CH_i without relay node	N.A.
L	Network lifetime	N.A.
$L_{CH_i}^{relay}$	lifetime for CH_i if it always transmits through the relay node	N.A.
$t_{CH_i}(j)$	time relay node serves CH_i in iteration j	N.A.
t_{CH_i}	total time relay node serves CH_i	N.A.
$t(j)$	time relay node serves CHs in iteration j	N.A.
$x(j)$	the improvement of network lifetime in iteration j	N.A.
$P_{RN, c}$	energy consumption rate for relay node	N.A.

We assume a TDMA frame (also denoted as a round) that comprises N_s slots: the $N_s - N_c$ NCH nodes will transmit to their corresponding CHs in the first $N_s - N_c$ slots while the CHs listens; the CHs will then transmit in the remaining N_c slots to BS, either directly or through the relay node. Through some message exchanges between the CHs and relay nodes, we assume that the relay node knows the residual energy in CH_i , E_{CH_i} , as well as the power consumption of CH_i with direct or indirect transmission to the BS, denoted by $P_{CH_i, on}$ and $P_{CH_i, off}$ respectively. Our problem is then to determine the amount of time the relay node

should serve CH_i , denoted by t_{CH_i} , to achieve maximum network lifetime, i.e., the time until the first CH dies. Table 4.1 summarizes the notations used throughout section 4.1.

Figure 4.1: Our proposed clustered WSN model.

4.1.2 Distributed Iterative Algorithm for Relay Node Scheduling

Let $L_{CH_i}^{direct}$ ($L_{CH_i}^{relay}$) denote the lifetime of CH_i if it always transmits *directly* (via the relay node) to the BS, i.e.,

$$L_{CH_i}^{direct} = \frac{E_{CH_i}}{P_{CH_i,off}} \quad (4.1)$$

$$L_{CH_i}^{relay} = \frac{E_{CH_i}}{P_{CH_i,on}} \quad (4.2)$$

If $L_{CH_i}^{direct}$ is sorted in increasing order, i.e., $L_{CH_1}^{direct} \leq L_{CH_2}^{direct} \leq L_{CH_3}^{direct} \dots \leq L_{CH_{N_c}}^{direct}$, then the network lifetime $L = L_{CH_1}^{direct}$. We propose a relay node scheduling algorithm that iteratively increments the lifetime of $\{CH_1, \dots, CH_{j-1}\}$ to match the lifetime of CH_j so that network lifetime can be maximized. Let $x(j) = L_{CH_{j+1}}^{direct} - L_{CH_j}^{direct}$, as illustrated in Figure 4.2. Let $t_{CH_i}(j)$ denote the duration that relay node serves CH_i , and $t(j) = \sum_{i=1}^j t_{CH_i}(j)$ denote the total time the relay node is serving CHs, in iteration j . We also define α_{CH_i} as follows:

$$\alpha_{CH_i} = \frac{P_{CH_i,off} - P_{CH_i,on}}{P_{CH_i,off}} \quad (4.3)$$

Since the relay node has initial energy E_s , it can operate for maximum duration of $\frac{E_s}{P_{RN,c}}$, where $P_{RN,c}$ is the rate of energy consumption when it is active.

Figure 4.2: Abstract model for lifetime expression.

• Iteration=1

We have the following theorem:

Theorem 4.1. *If $\frac{E_s}{P_{RN,c}}\alpha_{CH_1} \geq (L_{CH_2}^{direct} - L_{CH_1}^{direct})$ and $L_{CH_1}^{relay} > L_{CH_2}^{direct}$, then the relay node needs to serve CH_1 in iteration 1 for a duration $t_{CH_1}(1)$ so that its lifetime is $L_{CH_2}^{direct}$, where $\frac{x(1)}{t_{CH_1}(1)} = \alpha_{CH_1}$*

Proof. First, we have the following equation arrays:

$$\begin{cases} E_{CH_1} = L_{CH_1}^{direct} P_{CH_1,off} \\ E_{CH_1} = P_{CH_1,on} t_{CH_1}(1) + (L_{CH_1}^{direct} + x(1) - t_{CH_1}(1)) P_{CH_1,off} \end{cases} \quad (4.4)$$

According to (4.4), we have:

$$\frac{x(1)}{t_{CH_1}(1)} = \frac{P_{CH_1,off} - P_{CH_1,on}}{P_{CH_1,off}} \quad (4.5)$$

Therefore, Theorem 4.1 is proved. $\square$

Following this, the stored energy in the relay node is decremented by $P_{RN,c} \frac{L_{CH_2}^{direct} - L_{CH_1}^{direct}}{\alpha_{CH_1}}$.

• Iteration=2

The bottleneck nodes are now CH_1 and CH_2 , and their lifetime should next be improved (through relay node scheduling) until their lifetime is $L_{CH_3}^{direct}$. We have the following theorem:

Theorem 4.2. If $\frac{E_s}{P_{RN,c}}\alpha \geq (L_{CH_3}^{direct} - L_{CH_2}^{direct})$ where $\alpha = \frac{1}{\frac{1}{\alpha_{CH_1}} + \frac{1}{\alpha_{CH_2}}}$ and $L_{CH_i}^{relay} > L_{CH_3}^{direct} \forall i = (1, 2)$, then the relay node needs to serve CH_1 and CH_2 in iteration 2 for duration $t_{CH_1}(2)$ and $t_{CH_2}(2)$ respectively so that their lifetimes are $L_{CH_3}^{direct}$, where $\frac{x(2)}{t(2)} = \frac{1}{\frac{1}{\alpha_{CH_1}} + \frac{1}{\alpha_{CH_2}}}$.

Proof. For CH_1 , we have:

$$\begin{cases} E_{CH_1} &= P_{CH_1,on}t_{CH_1}(1) + (L_{CH_2}^{direct} - t_{CH_1}(1))P_{CH_1,off} \\ t_{CH_1}(1) &= \frac{L_{CH_2}^{direct} - L_{CH_1}^{direct}}{\alpha_{CH_1}} \\ E_{CH_1} &= (t_{CH_1}(1) + t_{CH_1}(2))P_{CH_1,on} + (L_{CH_2}^{direct} + x(2) - t_{CH_1}(1) - t_{CH_1}(2))P_{CH_1,off} \end{cases} \quad (4.6)$$

After proper transformation of (4.5), we have

$$\frac{x(2)}{t_{CH_1}(2)} = \frac{P_{CH_1,off} - P_{CH_1,on}}{P_{CH_1,off}} = \alpha_{CH_1} \quad (4.7)$$

Therefore, $\frac{x(2)}{t_{CH_1}(2)} = \alpha_{CH_1}$.

For CH_2 , we have the following equation arrays:

$$\begin{cases} E_{CH_2} &= L_{CH_2}^{direct}P_{CH_2} \\ E_{CH_2} &= P_{CH_2,on}t_{CH_2}(2) + (L_{CH_2}^{direct} + x(2) - t_{CH_2}(2))P_{CH_2,off} \end{cases} \quad (4.8)$$

After transformation of (4.8), we have the following (4.9).

$$\frac{x(2)}{t_{CH_2}(2)} = \frac{P_{CH_2,off} - P_{CH_2,on}}{P_{CH_2,off}} = \alpha_{CH_2} \quad (4.9)$$

Hence, $\frac{x(2)}{t_{CH_i}(2)} = \alpha_{CH_i}$ for $i = 1, 2$. Next, using (4.7) and (4.9), we derive the relationship between $t(2)$ and $x(2)$ as follows:

$$t(2) = t_{CH_1}(2) + t_{CH_2}(2) = \frac{x(2)}{\alpha_{CH_1}} + \frac{x(2)}{\alpha_{CH_2}} \quad (4.10)$$

$\Rightarrow$

$$\frac{x(2)}{t(2)} = \frac{1}{\frac{1}{\alpha_{CH_1}} + \frac{1}{\alpha_{CH_2}}} \quad (4.11)$$

□

Following this, the stored energy in the relay node is decremented by $P_{RN,c} \frac{L_{CH_3}^{direct} - L_{CH_2}^{direct}}{\alpha}$.

• **Iteration= j**

At iteration j , the relay node will serve CH_i , $1 \leq i \leq j$, until they have the same lifetime as CH_{j+1} . We have the following theorem:

Theorem 4.3. *If $\frac{E_s}{P_{RN,c}}\alpha \geq (L_{CH_{j+1}}^{direct} - L_{CH_j}^{direct})$ where $\alpha = \frac{1}{\sum_{i=1}^j \frac{1}{\alpha_{CH_i}}}$ and $L_{CH_i}^{relay} > L_{CH_{j+1}}^{direct} \forall i = (1, 2, \dots, j)$, then the relay node needs to serve $\{CH_1, \dots, CH_j\}$ in iteration j for duration $\{t_{CH_1}(j), \dots, t_{CH_j}(j)\}$ respectively so that their lifetimes are $L_{CH_{j+1}}^{direct}$, where $\frac{x(j)}{t(j)} = \frac{1}{\sum_{i=1}^j \frac{1}{\alpha_{CH_i}}}$*

Proof. For CH_i , $i=1,2,\dots,j$, we have the following equations:

$$\begin{cases} E_{CH_i} = \sum_{k=1}^{j-1} t_{CH_i}(k) P_{CH_i,on} + (L_{CH_j}^{direct} - \sum_{k=1}^{j-1} t_{CH_i}(k)) P_{CH_i,off} \\ E_{CH_i} = \sum_{k=1}^j t_{CH_i}(k) P_{CH_i,on} + (L_{CH_j}^{direct} + x(j) - \sum_{k=1}^j t_{CH_i}(k)) P_{CH_i,off} \end{cases} \quad (4.12)$$

From (4.12), we have the following equations:

$$\frac{x(j)}{t_{CH_i}(j)} = \alpha_{CH_i} \forall i = (1, 2, \dots, j) \quad (4.13)$$

The total time that the relay node needs to serve them is $t(j)$, which is shown below:

$$t(j) = \sum_{i=1}^j t_{CH_i}(j) = \sum_{i=1}^j \frac{x(j)}{\alpha_{CH_i}} \Rightarrow \frac{x(j)}{t(j)} = \frac{1}{\sum_{i=1}^j \frac{1}{\alpha_{CH_i}}} \quad (4.14)$$

□

In summary, our proposed relay node scheduling algorithm is shown in Algorithm 4.1. Note that if t_{CH_i} , the total time the relay node is scheduled to serve CH_i , is greater than the time slot duration, T , then it will serve CH_i for several rounds N until $NT \leq t_{CH_i}$ and $(N+1)T > t_{CH_i}$; CH_i will transmit directly to the BS in subsequent rounds. The algorithm is found to converge as it will terminate once $L < L_{CH_{j+1}}^{direct}$.

So far we developed the distributed clustering algorithm with single battery powered relay node. Next, we extend the algorithm to multiple energy harvesting relay nodes.

Algorithm 4.1 Algorithm for relay node to re-select CH

- **Initialization**

Relay node collects E_{CH_i} , $P_{CH_i,off}$, $P_{CH_i,on}$ from CH_i and $P_{RN,c}$. Relay node sorts $L_{CH_i}^{direct}$ in increasing order according to (4.1) such that $L_{CH_1}^{direct} \leq L_{CH_2}^{direct} \dots \leq L_{CH_{N_c}}^{direct}$. It calculates α_{CH_i} and $L_{CH_i}^{relay}$ for CH_i according to (4.3) and (4.2). Set $j = 1$.

- **Step 1**

If $L_{CH_i}^{relay} > L_{CH_{j+1}}^{direct}$ for $1 \leq i \leq j$, goto Step 2, else goto step 4.

- **Step 2**

Let $\alpha = \frac{1}{\sum_{i=1}^j \frac{1}{\alpha_{CH_i}}}$. If $(L_{CH_{j+1}}^{direct} - L_{CH_j}^{direct}) > \frac{E_s}{P_{RN,c}} \alpha$ Then go to step 4; else, goto Step 3.

- **Step 3**

Record $t_{CH_i}(j) = \frac{L_{CH_{j+1}}^{direct} - L_{CH_j}^{direct}}{\alpha_{CH_i}}$. Update $E_s = E_s - P_{RN,c} \frac{L_{CH_{j+1}}^{direct} - L_{CH_j}^{direct}}{\alpha}$. Set $j = j + 1$. Return to Step 1.

- **Step 4**

The network lifetime is thus $L = L_{CH_j}^{direct} + \frac{E_s}{P_{RN,c}} \alpha$. Also, $t_{CH_i} = \frac{L - L_{CH_i}^{direct}}{\alpha_{CH_i}}$.

4.2 Distributed Clustering Algorithms for Multiple-clustered WSN with Multiple Energy Harvesting Relay Nodes

In this section, we design the distributed clustering algorithm to include multiple energy harvesting relay nodes. Specifically, we propose the network model in section 4.2.1. Then we propose the distributed algorithm in section 4.2.2. Lastly, we also propose some benchmark algorithms to compare with in section 4.2.3.

4.2.1 Network Model

The network scenario we adopt in section 4.2 is shown in Figure 4.3. Without loss of generality, we make the following assumptions in the model throughout section 4.2:

- We assume a large scale network with N_s sensors with the same initial energy and N_e EH nodes randomly deployed in fixed locations within a square region with lower left hand

vertex (a, b) and dimension M . The sensor nodes are partitioned into N_c clusters, each comprising one CH. Each EH serves as the relay for one CH or multiple CHs, which can be different at different times. A typical EH node's energy profile is shown in Figure 4.4 where it transmits data during the *on* phase, goes to *off* phase to harvest energy from the environment and repeats the cycle after it has harvested enough energy. The BS is deployed at $(0,0)$.

Figure 4.3: Abstract model of multiple EH nodes serving as relay for CHs.

- The distributed network operation in each round assumes TDMA and CSMA- based communication [13] and comprises 2 phases, namely, the setup phase and transmission phase, as shown in Figure 4.5. In the setup phase, selection of CHs, formation of clusters and matching between EHs and CHs are performed. We assume that sufficient time is allocated in the setup phase for these operations, and that each CH and EH node consume overhead of a few bytes (assumed to be 4 bytes in the simulation which is

Figure 4.4: Typical energy profile of EH node with time.

Figure 4.5: Overall flow chart for our design.

enough for signaling). Each node can only make decisions based on local information such as residual energy and transmission power of nearby nodes. During the transmission phase, each NCH transmits a packet of 2000 bits to its respective CH in a TDMA time slot assigned by the latter during the setup phase. Then CHs forward data to BS using CSMA approach¹, either directly, or via EH node. The network lifetime is measured in number of TDMA rounds the network can operate until the first node dies.

¹We assume the collision of transmissions does not have significant impact on network lifetime, similarly as assumed in [13].

- We assume the Friss free-space propagation model [13], where the transmission power is proportional to the distance square between sensors. Note that the proposed algorithm can be easily extended to other propagation models, e.g., multi-path fading model [118].

The notations used in section 4.2 are shown in Table 4.2.

Table 4.2: Notations used throughout section 4.2

Notation	Description	Value
(a, b)	Lower left hand vertex of deployed region	N.A.
M	Dimension of square region	N.A.
N_s	Number of sensors in the network	$\{100, 125, 150, 175, 200\}$
N_e	Number of EH sensors in the network	N.A.
N_c	Number of clusters in the network	N.A.
i	Cluster index	$\{i=1, 2, \dots, N_c\}$
j	EH index	$\{j=1, 2, \dots, N_e\}$
T	Time duration for each round	N.A.
t_r	Length of re-clustering period	N.A.
E_s	Energy stored in battery for each sensor	0.5J
CH_i	The CH node for cluster i	N.A.
E_{CH_i}	Residual energy in CH_i	N.A.
E_{th}	Energy threshold for node to wake up	N.A.
$P_{EH,h}$	Energy harvesting rate for EH node	N.A.
$P_{CH_i,BS}$	Transmission power between CH_i and BS	N.A.
P_{CH_i,EH_j}	Transmission power between CH_i and EH_j	N.A.

4.2.2 Distributed Clustering Algorithm for Maximizing Lifetime in WSNs with EH Sensors

Similarly as that proposed in [13], our clustering algorithm also re-selects CH periodically at the beginning of every round. We extend the work in [13] by considering energy harvesting sensors as relay for CHs. Specifically, in Section 4.2.2.1, we propose cluster head selection algorithm; followed by the cluster formation algorithm in Section 4.2.2.2. Lastly we show our matching algorithm in Section 4.2.2.3.

4.2.2.1 Cluster Head Selection Algorithm

In the setup phase, sensors nominate themselves to serve as CH. Once the sensor is determined to be CH, it broadcasts an *announcement* message within its range. Each sensor has a probability of p_{s_i} to serve as CH. The expected number of CHs is $N_s p_{s_i}$. We show the optimal number of CHs, denoted as $N_c^{opt} = N_s p_{s_i}^{opt}$ through simulations in Section 4.3.2.1.

4.2.2.2 Cluster Formation Algorithm

Similarly as that in [13], in this step, every sensor selects the closest CH to join. The distance of a CH is indicated by the strength of signal received by NCHs. The closest CH from each NCH is the one with strongest signal. Once NCHs determine the closest CH, they send *join* message to it.

4.2.2.3 Matching Algorithm

After cluster head selection and cluster formation, our next step is to design the matching algorithm between CHs & EHs to maximize the time until the first CH dies. We consider the scenario when each EH node can serve as relay for only one CH in each round. We extend the consideration to the case when each EH node can serve as relay for multiple CHs in a round in Section 4.2.3.2.

We assume that every sensor including CH or EH makes decision based on its local information collected.

The main idea of our algorithm is that through several iterations, the algorithm can find a matching between CHs and EHs such that the CH node that EH node needs to serve is the sensor with lowest residual energy that has not been served by other EHs. By serving the sensor with lowest residual energy every round, the number of rounds till the first CH dies can be increased. For the CH, the best EH node that it could find is the *closest* EH node in its neighborhood, followed by the second closest and so on.

In order to calculate the residual energy of CHs, we design the following approaches. We use $E_{CH_i}^{off}$ to denote the residual energy of CH_i if it communicates directly with BS. E_{CH_i} denotes the residual energy of CH_i at the beginning of a round. Let $E_{CH_i}^{on}$ denote the residual energy in CH_i after each round with EH node serving as relay for CHs, and denote the residual

energy for CH_i after a round if it directly communicates with EH_j for a whole round as E_{CH_i, EH_j}^{on} .

After a round (duration T), $E_{CH_i}^{off}$ is shown in the following equation:

$$E_{CH_i}^{off} = E_{CH_i} - P_{CH_i, BS}T \quad (4.15)$$

Let t_{ij} represent the time EH_j serves as relay for CH_i in each round. In our scheme, EH node serves as relay for CH for a whole round, hence if $t_{i^*j^*} = T$ for CH_{i^*} and EH_{j^*} , then $t_{ij} = 0$ for $i = i^*, j \neq j^*$ and $i \neq i^*, j = j^*$. For simplicity of the scheme design, we assume EH node can serve as relay for CHs only if its residual energy can sustain transmission for at least one round with duration T . Besides, we assume that, during the transmission, EH nodes that wake up do not broadcast *awake* messages until the beginning of the next round.² We have:

$$E_{CH_i} - E_{CH_i}^{on} = (T - \sum_{j=1}^{N_e} t_{ij})P_{CH_i, BS} + \sum_{j=1}^{N_e} t_{ij}P_{CH_i, EH_j} \quad (4.16)$$

$\Rightarrow$

$$E_{CH_i}^{on} = E_{CH_i} - P_{CH_i, BS}T + \sum_{j=1}^{N_e} t_{ij}(P_{CH_i, BS} - P_{CH_i, EH_j}) \quad (4.17)$$

Let

$$\alpha_{ij} = P_{CH_i, BS} - P_{CH_i, EH_j}. \quad (4.18)$$

From (4.15), (4.16), (4.17), (4.18), we have:

$$E_{CH_i}^{on} = E_{CH_i}^{off} + \sum_{j=1}^{N_e} t_{ij}\alpha_{ij}.$$

Specifically, E_{CH_i, EH_j}^{on} is expressed as follows:

$$E_{CH_i, EH_j}^{on} = E_{CH_i}^{off} + T\alpha_{ij} \quad (4.19)$$

At setup phase, we use Algorithm 4.2 to match CHs and EHs. It is important to analyze the number of iterations to converge and the performance of algorithm. While a near optimal

²Since EHs may wake up during the transmission, they may broadcast *awake* message to their surrounding CHs. If they do so, however, CHs may consume quite a lot of energy to re-match the new EHs. For simplicity, when EHs wake up during the transmission, they will wait until the beginning of the next round to broadcast *awake* message.

Algorithm 4.2 Distributed matching algorithm at setup phase

Require: N_c CHs and N_e EHs**Ensure:** Matching between CHs and EHs

At the beginning, EH node broadcasts *awake* message to CHs in their broadcast range. Each CH estimates P_{CH_i, EH_j} according to the signal strength and calculates E_{CH_i, EH_j}^{on} . It also records the smallest $E_i = \min E_{CH_i, EH_j}^{on}$.

while there are unmatched CHs **do** *For each unmatched CH:* **if** there are unmatched EH nodes **then** It selects the *closest* unmatched EH node and send *request* message including E_i to this EH node. **else**[CH node does not receive any *awake* message] It communicates directly with *BS*. **end if** *For each unmatched EH:* EH node receives several *request* messages from CH nodes in its range. Denote the number of *requests* as N_r , N_r may be equal to 0 **if** $N_r > 0$ **then** EH node sends *grant* to the CH node with smallest E_i . **else**[$N_r = 0$]

EH node waits for the next iteration.

end if *For each unmatched CH:* CH node may or may not receive *grant* messages from EH nodes in its range, let N_g be the number of *grant* messages it receives. **if** $N_g = 1$ **then** It selects the closest EH node and sends *accept* message to it. **else** $N_g = 0$

Return to while loop.

end if **end while**

matching between CHs and EHs can be achieved through many iterations of signaling between CHs and EHs, this consumes higher energy than the overhead. Hence there is a tradeoff between the performance of the algorithm and number of iterations/signals that each EH/CH may spend for the matching.

Since in every iteration, at least one CH is matched to either EH or BS, Algorithm 4.2 requires at most N iterations to converge, where $N = \min(N_c, N_e)$.

Figure 4.6 shows the average number of iterations required for convergence while 100

simulations are performed for each case. The same number of CHs and EH nodes are randomly deployed in a $100\text{-}200\text{ m}^2$ region. We change the number of CHs and EHs from 6 to 55. The simulation result shows that the process will converge in approximately $\mathcal{O}(\log_2(N))$ iterations on average.

Figure 4.6: The average numbers of iterations required to converge while using the Algorithm 4.2 in WSNs with different numbers of EHs/CHs.

Figure 4.7: Timing diagram for message exchanges in overhead.

We use the timing diagram in Figure 4.7 to show the overhead messages exchanged during the setup phase.

4.2.3 Benchmark Matching Algorithms for Maximizing Lifetime of WSNs with EH Sensors

Algorithm 4.2 in Section 4.2.2.3 provides distributed matching between CHs and EHs to maximize lifetime. In order to show the performance of our algorithm, we propose centralized benchmark algorithms using information about all CHs and EHs in this section. Specifically, in Section 4.2.3.1, we propose graph-based benchmark which serves as upper bound for Algorithm 4.2. In Section 4.2.3.2, we define LP-based benchmark.

At the beginning of setup phase, after some information exchanges between CHs and EHs (i.e., the overhead shown in Figure 4.7 that we take into consideration), we assume that:

- N_e EH nodes know the residual energy and energy consumption rate of each CH.
- The energy consumed in the overhead occurs once at the beginning of each round.

4.2.3.1 Graph-based Matching Algorithm

The objective for the algorithm is to find a matching between CHs and EHs or BS in order to maximize the minimal E_{CH_i, EH_j}^{on} or $E_{CH_i}^{off}$ (defined in Section 4.2.2.3). In other words, the problem is similar to a max-min matching problem in bipartite graph. CH represents one set in a bipartite graph while EH and BS represent the other set. The problem is to look for the "largest set of edges" that cannot be removed. A *matching* exists if all CHs have been matched to either EH node or BS. Each EH node can only match one CH node while BS can match multiple CH nodes. Each CH node can only be matched to one EH node or BS.

Our idea is to assign proper weights to edges between both sets taking into consideration that BS can be matched to multiple CHs. Then we sort these edges in an ascending order and use binary search and Hopcroft-Karp Algorithm [121] to find the set of edges that cannot be removed.

Let $W(i, j)$ be the weight assigned to the edge between CH_i and EH_j or BS . First we derive an $N_c(N_e + N_c)$ matrix W with row index $1 \leq i \leq N_c$ and column index $1 \leq j \leq N_e + N_c$. $W(i, j)$ in row i and column j where $1 \leq j \leq N_e$ represents E_{CH_i, EH_j}^{on} according to (4.19). $W(i, j)$ in row i and column j where $N_e + 1 \leq j \leq N_e + N_c$ represents $E_{CH_i}^{off}$ according to (4.15).

Algorithm 4.3 Optimized Graph-based Algorithm for multiple EH nodes to re-select CH as relay at setup phase

• **Step 1**

Sort $W(i, j)$ in ascending order, where $1 \leq i \leq N_c$ and $1 \leq j \leq N_e + N_c$. We record the sorted array as L_i , where $1 \leq i \leq N_c(N_e + N_c)$. We have $L_1 \leq L_2 \dots \leq L_{N_c(N_e + N_c)}$. Let iteration $k = 1$, define $lb = 1, ub = N_c(N_e + N_c)$.

• **Step 2**

Let $b_k = \lfloor \frac{lb+ub}{2} \rfloor$. Use the Hopcroft-Karp Algorithm [121] to check if there still exists a matching if we remove all L_i for $i \leq b_k$. If yes, remove all the L_i for $i \leq b_k$ and let $lb = b_k$; otherwise, do not remove any element but let $ub = b_k$.

• **Step 3**

Let $k = k + 1$, repeat Step 2 until $lb = ub$. We generate the matches between corresponding CH_i and EH_j and record L_{lb} .

We sort $W(i, j)$ in an ascending order and then remove the smallest $W(i, j)$ one by one until we cannot find a *matching*. In our case, a *matching* exists if the maximal number of matches is larger than or equal to N_c . By adopting binary search to cut the "largest" edge and check if there is a matching, the time complexity will be reduced from $O(n^4\sqrt{n})$ to $O(n^2\sqrt{n}\log(n))$. The algorithm is shown in Algorithm 4.3³.

The optimality of the algorithm is defined as whether it can generate a matching such that within the matching, $\min(W(i, j))$ can be maximized in each round. We prove that Algorithm 4.3 can maximize $\min(W(i, j))$ in a *matching* for each round in Theorem 4.4.

Theorem 4.4. *Algorithm 4.3 can maximize $\min(W(i, j))$ in a matching for each round.*

Proof. Proof by contradiction: suppose there is an optimal *matching* denoted as W^* where CHs have been matched to either EHs or BS. Let $L^* = \min(W^*(i, j)), 1 \leq i \leq N_c, 1 \leq j \leq N_e + N_c$. We denote the *matching* from Algorithm 4.3 as $\overline{W}(i, j), 1 \leq i \leq N_c, 1 \leq j \leq N_e + N_c$. Let $\overline{L} = \min(\overline{W}(i, j)) = L_{lb} = L_{ub}$.

³There are many possible solutions to maximize the minimal E_{CH_i, EH_j}^{on} or $E_{CH_i}^{off}$. Our scheme can find one solution among them. The algorithm can be extended to maximize the second minimal E_{CH_i, EH_j}^{on} or $E_{CH_i}^{off}$, the third minimal E_{CH_i, EH_j}^{on} or $E_{CH_i}^{off}$ and so on, which is out of the scope of this thesis.

We have $L^* > \bar{L}$, which means *matching* W^* exists for L_i when $i > lb$. However, according to Algorithm 4.3, there is no *matching* for L_i when $i > lb$, which is a contradiction. $\square$

4.2.3.2 LP-based Matching Algorithm

In our previous analysis, we considered a scenario when each EH serves as relay for one CH in each round, which was denoted as Scenario A. In this section, we consider a new scenario, denoted as scenario B, where each EH node can serve as relay for multiple CHs in a round. Intuitively, Scenario B can achieve better lifetime compared with Scenario A if overhead is small. Scenario A is simple for a large scale network and has lower control overhead compared with Scenario B.⁴

LP approach is a benchmark algorithm for Scenario B. It provides longer lifetime compared with Algorithm 4.3 for Scenario A when the control overhead is ignored. We compare these two different scenarios and see the improvements Scenario B can have over Scenario A.

The problem can be formulated as follows: we would like to find t_{ij} for $i=1,2,\dots,N_c$ and $j=1,2,\dots,N_e$ such that the minimum residual energy left in CH_i is maximized after each round. Quantitatively, this can be expressed as the following max-min problem:

MMWSN: To maximize $z = \min(E_{CH_i}^{on} = E_{CH_i}^{off} + \sum_{j=1}^{N_e} t_{ij}\alpha_{ij}) \forall i$

s.t.:

$$(1) \sum_{j=1}^{N_e} t_{ij} \leq T \forall i$$

$$(2) \sum_{j=1}^{N_e} t_{ij} P_{CH_i, EH_j} \leq E_{CH_i}$$

$$(3) t_{ij} \geq 0, \forall i, \forall j$$

Note that $E_{CH_i}^{on} = E_{CH_i}^{off}$ if $t_{ij} = 0$ for all j . $\sum_{j=1}^{N_e} t_{ij}$ in constraint (1) represents the total time CH_i has been served by all EHs in a round. Since the duration of each round is T , constraint (1) ensures that the total time all EH_j serve as relay for each CH cannot exceed T . Constraint (2) ensures that the energy consumption by CH_i does not exceed its residual energy. Constraint (3) makes sure that t_{ij} is not negative. This max-min optimization problem is similar to the problem [122], denoted by MM, of allocating resources $j \in 1, \dots, J$ to activities $k \in 1, \dots, K$, as shown below:

⁴Distributed algorithm for scenario A can be extended to suit scenario B with more overhead and higher complexity. We show in Section 4.3 that, under our deployment cases, the optimal lifetime for Scenario A and Scenario B has slight difference. Thus we only derive the centralized algorithm for Scenario B in this section.

MM: To maximize $z = \min_i [r_i(\sum_{j=1}^J \beta_{ij} x_{ij})]$

s.t.:

$$(1) \sum_{i=1}^K x_{ij} = h_j, j = 1, 2, \dots, J$$

$$(2) x_{ij} \geq 0$$

x_{ij} is the quantity of resource j allocated to activity i ; h_j is the available quantity of resource j ; β_{ij} is the effectiveness of resource j when allocated to activity i ; and r_i is a return function for activity i . Specifically, $r_i(x_{ij}) = a_i + b_i(\sum_{j=1}^J \beta_{ij} x_{ij})$.

By letting $r_i = E_{CH_i}^{off} + \sum_{j=1}^{N_e} t_{ij} \alpha_{ij}$, $x_{ij} = t_{ij}$, $h_j = T$, **MMWSN** is similar to **MM** for both objective and constraints.

According to [122], an optimal solution of this problem can be obtained by solving the following linear programming problem:

MMWSN-LP: To maximize z

s.t.:

$$(1) E_{CH_i}^{off} + \sum_{j=1}^{N_e} t_{ij} \alpha_{ij} = E_{CH_i}^{on} \geq z \forall i$$

$$(2) \sum_{j=1}^{N_e} t_{ij} \leq T \forall i$$

$$(4) \sum_{j=1}^{N_e} t_{ij} P_{CH_i, EH_j} \leq E_{CH_i}$$

$$(5) t_{ij} \geq 0, \forall i, \forall j$$

Since the constraints are linear, this **MMWSN-LP** can be optimally solved according to [122]. We describe the algorithm in Algorithm 4.4.

4.3 Simulation Results

In this section, we show the simulation results for both algorithms derived in section 4.1 and section 4.2. Specifically, in section 4.3.1, we show the performance of clustering algorithms with single battery powered relay node. In section 4.3.2, we show the performance of clustering algorithm with multiple energy harvesting relay nodes.

4.3.1 Performance of Distributed Clustering Algorithms with Single Battery Powered Relay

We validate our results through simulations for clustered WSN with 10 clusters, each composing 20-30 sensor nodes. In each round, each node needs to transmit a 2000-bit packet at

Algorithm 4.4 LP Algorithm for multiple EH nodes to re-select CH as relay at setup phase

- **Step 1**

Calculate $E_{CH_i}^{off}$ and α_{ij} for CH_i and EH_j according to (4.15) and (4.18).

- **Step 2**

Formulate the problem as Max-min-sum problem as shown in **MMWSN-LP**. After solving this problem, we use t_{ij} to represent the optimal time EH_j serves as relay for CH_i .

- **Step 3**

Repeat Steps 1-2 at the beginning of each round until one of the CHs has $E_{CH_i} < 0$. Finally, we record the network lifetime in terms of number of rounds.

a rate of 250 kbps. We set the slot duration to be $T=0.016s$, $E_s = 0.5J$, and assume the Friss free space propagation model. We benchmark our proposed relay scheduling algorithm with (i) a periodic algorithm and (ii) direct CH-BS transmission. In the periodic algorithm, the relay node exchanges information with the CH nodes in the beginning of each round, and serves the CH with the lowest lifetime in that round.

We consider two square regions of network deployment, where the coordinates of the vertices are as follows: Area (I) (100, 100), (100, 200), (200, 100) and (200, 200); Area (II) (100, -50), (100, 50), (200, -50), (200, 50). The relay node is deployed at six different positions as follows: Case (I) (50,50); Case (II) (60,60); Case (III) (70,70); Case (IV) (50,0); Case (V) (60,0); Case (VI) (70,0);

We plot the time until each CH runs out of energy for each algorithm for {Case (I), Area (I)} and {Case (II), Area (I)} in Figure 4.8.a and 4.8.b respectively. As expected, direct transmission to the BS results in the worst network lifetime (10 rounds). The periodic approach increases the lifetime of CH_1 to CH_3 , resulting in a network lifetime of 20 rounds. While our proposed approach is more optimized than the periodic approach, it has lower overhead as it does not require message exchange in every round, resulting in a significant gain in network lifetime. The network lifetime is higher in Case II as the relay node was deployed closer to the WSN, resulting in lower power consumption by the CHs.

Next, we compare the network lifetime (i.e., time till the first CH node dies) obtained with each algorithm for all six cases in Figure 4.9.a. While our proposed approach achieves the

4.8.a

4.8.b

Figure 4.8: Number of survival rounds of each CH for (a) Case I and Area I (b) Case II and Area I

4.9.a

4.9.b

Figure 4.9: (a) Comparison of network lifetime for six different cases (b) Number of rounds relay node serves for each CH

highest network lifetime in all cases, the periodic approach yields better performance than direct CH-BS transmission except for {Case (V), Area (II)} and {Case (VI), Area (II)}. This is because the energy savings by transmitting via the relay node is smaller than the cost of energy overhead for relay node reselection in each round. Amongst the various scenarios, our approach achieves the best performance for {Case (II), Area (I)} and {Case (VI), Area (II)}. This can be explained as follows: In Case (II), the relay node can serve all CHs for a longer time (163.38 rounds) than in Case (III) (121.38 rounds) since the relay node is closer to the BS in Case (II) than in Case (III) which leads to a smaller energy consumption. For Case (I), CH_1 is farther from the relay node than in Case (II), which results in a shorter lifetime (27 rounds compared to 30.56 rounds in Case (II)).

Lastly, we plot the number of rounds the relay node serves each CH with our proposed approach and the periodic approach for $\{\text{Case (II), Area (I)}\}$ in Figure 4.9.b. We find that the relay node serves all CHs under our approach while it only serves the first two CHs under the periodic approach. The lifetime for CH_1 is improved by 15 rounds for our approach compared with periodic approach. This can be explained as follows: for the periodic approach, 0.015 J energy is spent on re-selection of CHs, which reduces the number of rounds it serves CH_1 by 15 compared with the optimal approach.

4.3.2 Performance of Distributed Clustering Algorithms with Multiple Energy Harvesting Relay Nodes

We demonstrate the performance of our proposed algorithms through simulations using MatLab for a 2-D network with a base station deployed at (0,0) and $N_s = \{100, 125, 150, 175, 200\}$ nodes randomly distributed over a square region, each initially equipped with 0.5 J of energy and has a data transmission rate of 2000 bits per packet. We consider three square regions of network deployment, where: **Case I** $\{a = 100, b = 100, M = 100\}$; **Case II** $\{a = 200, b = 200, M = 100\}$; **Case III** $\{a = 0, b = 0, M = 200\}$. Although Cases I and II have the same area, nodes are located closer to the BS in Case I than Case II. Cases I and III differ in terms of node density. By using these three cases, we can determine the influence of proximity to the BS and node density on the performance of our proposed algorithms.

The performance metric used in this thesis is the number of rounds the network could operate until the first node dies. We first show the effects of parameters $\{N_c, t_r, N_e, P_{eh,h}\}$ on the network lifetime in Section 4.3.2.1. We show the effects of parameters in Case I only, the same conclusions also hold for Case II and Case III. $P_{eh,h}$ is assumed according to the previous literature [4, 120], typically tens of milliwatts. Then we verify the sub-optimality of our algorithm in Section 4.3.2.2. We benchmark our distributed algorithm with algorithms proposed in Section 4.2.3.1 and Section 4.2.3.2. We also compare our results with PIM shown in Section 2.1.3.

4.3.2.1 Performance Characteristics of Distributed Algorithm [Case I]

We first examine the effects of number of clusters on network lifetime when $N_e = 4$, $P_{eh,h} = 0.04W$, $t_r = T$ and $E_{th} = E_s$. We plot the results from our simulation in Figure 4.10 when $1 \leq N_c \leq 10$ for Case I. The results show that the network lifetime is longest when $N_c^{opt} = 4$.

The effects of t_r on the algorithm performance in Case I are shown in Figure 4.11 when $p_{s_i} = 0.04$, $N_e = 4$, $P_{eh,h} = 0.04W$ and $E_{th} = E_s$. The results, obtained by averaging over 20 runs, show that the length of re-clustering period have effects on the lifetime. Specifically, in Case I, when p_{s_i} is varied between $\{0.02, 0.025, 0.03\}$, the optimal length of re-clustering period also varies. Specifically, the optimal length of re-clustering period is $2T$, $3T$ and $2T$ for these three different p_{s_i} , respectively. When $p_{s_i} = 0.025$, the lifetime when $t_r = 3T$ is 3.7% longer than the lifetime when $t_r = T$. This is because the more frequently CHs are re-selected, the higher the energy consumed. It is interesting, though not a big surprise, to see that when the length of re-clustering period increases, network lifetime increases first and then decreases. When re-clustering period increases, those CHs which were not *bottleneck* CHs previously may become *bottleneck* CHs since until the next re-clustering, they might have to consume high transmission power to transmit to farther EHs. Therefore, the network lifetime decreases due to the high power consumption over long time by these new *bottleneck* CHs.

Figure 4.10: Effects of number of clusters on network lifetime in Case I ($N_e = 4$, $P_{eh,h} = 0.04W$, $t_r = T$, $E_{th} = E_s$).

Note that we have four EH nodes in the previous few figures. Now we show the effect of the number of EH nodes when $p_{s_i} = 0.04$, $P_{eh,h} = 0.04W$, $t_r = T$ and $E_{th} = E_s$ in Figure 4.12. As expected, when the number of EH nodes increases, network lifetime increases. Interestingly, when N_e is increased further, i.e., $N_e > 10$, the increase in the network lifetime becomes less and less significant. With further increase of N_e , the network lifetime will not increase. Theoretically, when N_e approaches infinity, CHs will not become bottleneck nodes.

Instead, NCHs which are farthest from CHs will become bottleneck nodes, thus affecting the network lifetime.

Figure 4.11: Effects of t_r on network lifetime in Case I ($p_{s_i} = 0.04$, $N_e = 4$, $P_{eh,h} = 0.04W$, $E_{th} = E_s$)

Figure 4.12: Effects of number of EH nodes on the network lifetime in Case I ($p_{s_i} = 0.04$, $P_{eh,h} = 0.04W$, $t_r = T$, $E_{th} = E_s$).

We also show the effects of EH harvesting rate on network lifetime in Figure 4.13 when $N_e = 4$, $p_{s_i} = 0.04$, $t_r = T$ and $E_{th} = E_s$. The EH harvesting rate is increased from 0.01 W to 0.1 W in steps of 0.01 W. When EH harvesting rate is 0, the case simplifies to battery powered network, already shown in section 3.5. As expected, when harvesting rate increases,

Figure 4.13: Effects of EH node harvesting rate on the network lifetime in Case I ($N_e = 4$, $p_{s_i} = 0.04$, $t_r = T$, $E_{th} = E_s$).

network lifetime increases as well. Specifically, when harvesting rate increases from 0.01 W to 0.04 W for 100 sensors, network lifetime increases by 44.4% from 432.55 to 624.65. Interestingly, we also notice that when harvesting rate continues to increase, the increase in lifetime becomes less and less significant; with further increase of harvesting rate, network lifetime will approach constant. Theoretically when harvesting rate approaches infinity, CHs will not become bottleneck nodes. Instead, NCHs which are farthest from CHs will become bottleneck nodes, thus affecting the network lifetime.

Then we illustrate the performance of the proposed algorithms for different E_{th} ranging from $\{0.1E_s, 0.2E_s, \dots, E_s\}$ for case I when $N_e = 4$, $p_{s_i} = 0.04$, $t_r = T$ and $P_{eh,h} = 0.04W$. We compare the performance of our algorithm when number of sensors is 100 and 150 for Case I, and plot the results in Figure 4.14. As seen from this figure, the network lifetime is not a monotonic function of E_{th} . For example, the optimal lifetime is achieved when remaining energy is $0.7E_s$ when number of sensors is 100, and is $0.6E_s$ when number of sensors is 150. As shown in Section 4.2.2.3, each EH node broadcasts *awake* message when they can transmit for the whole round. From the figure, we can see that E_{th} does not significantly affect the network lifetime. For example, the increase in the lifetime when E_{th} is $0.7E_s$ is only 3.16% more than the case when E_{th} is E_s when $N_s = 100$. When $E_{th} = E_s$, the time EH node takes to recharge is longer compared with the time for EH node to recharge when $E_{th} = 0.7E_s$. On

the other hand, the time the EH node serves as a relay for CHs when $E_{th} = E_s$ is also longer compared with the time when $E_{th} = 0.7E_s$.

Figure 4.14: Effects of E_{th} on the network lifetime in Case I ($N_e = 4$, $p_{s_i} = 0.04$, $t_r = T$, $P_{eh,h} = 0.04W$).

4.3.2.2 Comparison of Distributed Algorithm with Benchmark Algorithms

Next, we compare the performance of our distributed algorithm (DA in short) with respect to Graph-based benchmark (GB in short) (shown in Section 4.2.3.1), LP-based Benchmark (LP in short) (shown in Section 4.2.3.2) and PIM (random matching shown in Section 2.1.3). We adopt the same parameters in these approaches, that is $p_{s_i} = 0.04$, $N_e = 4$, $t_r = T$, $P_{eh,h} = 0.04W$ and $E_{th} = E_s$, which are selected according to Section 4.3.2.1.

For Case I, network lifetime increases when N_s increases as shown in Figure 4.15. The reason is that when number of sensors increases, more sensors will be able to serve as CHs, which reduces the chances that CHs become *bottleneck* nodes. The result shows that our algorithm outperforms PIM and is quite close to the benchmarks. Specifically, for Case I with 100 and 150 sensors, the network lifetime from DA is 63.5% and 48.8% longer compared with PIM. This is because our algorithm selects the EH node to serve the best CHs it could find; while PIM selects EH nodes to randomly serve a certain CH. Thus, our algorithm maximizes the lifetime until the first node dies while PIM is trying to find a matching within fewer iterations.

Figure 4.15: Comparison between our algorithm with benchmark algorithms in Case I ($p_{s_i} = 0.04$, $N_e = 4$, $t_r = T$, $P_{eh,h} = 0.04W$, $E_{th} = E_s$).

When we compare DA with the benchmark algorithms, we found the network lifetime in Case I for DA with 100 and 150 sensors can achieve 93.4%, 94.4% and 92.1%, 92.0% optimality compared with LP and GB. Note that⁵:

- (i) We are optimizing the performance within a few iterations while LP and GB adopt more complicated calculations.
- (ii) Our algorithm is based on the local information that each EH or CH node receives while the benchmark algorithms uses global information to make decisions.

Hence we may claim that the proposed algorithm achieve satisfactory performance and good scalability.

As mentioned in Section 4.2.3.2, we compare the Scenario A and Scenario B in this section. GB and DA are for Scenario A while LP is for Scenario B. The result shows that LP only provides slightly longer lifetime compared with GB. For example, with 100 and 150 sensors, the differences between LP and GB results is 1% and 0.1% respectively. The differences between Scenario A and Scenario B is small in Case I, however, for some special deployment scenarios and parameters, the difference may be large, which is not covered in this work.

⁵Strictly speaking, these benchmarks only provide optimized results for each round rather than optimizing the overall network lifetime. Optimizing overall network lifetime requires much more complicated approaches, which is out of the scope of our work.

4.4 Conclusion

In this chapter, we considered clustered wireless sensor networks (WSN) where CHs either aggregate and forward data directly to BS, or via dedicated relay nodes with/without energy harvesting (EH) capabilities. We proposed an optimal distributed algorithm for scheduling a single battery powered relay node to serve the cluster heads to maximize network lifetime. We also proposed efficient distributed matching algorithm between CHs and multiple EH relay nodes to maximize network lifetime, where the network lifetime is the duration until the first node runs out of energy. Through theoretical analysis and extensive simulations, we validated the performance of the proposed algorithms. Simulation results demonstrate much better performance (around 63.55% longer lifetime) compared with randomized approach.

So far we have developed both centralized and distributed lifetime maximization algorithms. In the next chapter, we shall study the event detection problem.

Chapter 5

Event Detection Algorithms in Random WSNs

This chapter presents event detection algorithms for random wireless sensor networks, including both Nonhomogenous Binomial Point Process (NBPP) and Infinite Poisson Point Process (IPPP) models. Specifically, Section 5.1 shows the system model and problem formulation for the event detection problem. Section 5.2 shows the event detection algorithm under random sensor deployment. Section 5.3 shows the probability density estimation with series expansion methods. In Section 5.4, we calculate the moments required by the series expansion methods. In Section 5.5, extensive simulation results and discussions are presented for verifying the performance of the proposed algorithms. Section 5.6 concludes this chapter.

5.1 System Model and Problem Formulation

In this section, a formal definition of the Finite Binomial Point Process and Infinite Poisson Point Process is presented first, followed by system model assumptions. These processes are special cases of spatial point process. In general a spatial point process is a random pattern of points, in our case in 2-dimensional space.

One of the most useful ways to handle a spatial point process is to generalise the notion of one-dimensional spatial point processes involving interval counts $N(a, b]$ to the concept of region counts $N(B)$ which is the number of points falling in a set $B \subset \Omega \subseteq \mathbb{R}^2$. One may then characterize a spatial point process by two measures, the counts of points in sets, i.e., sets where $N(B) > 0$ for regions B and the vacancy sets $V(\tilde{B}) = N(\tilde{B}) = 0$ where there are

no counts present. The two most commonly used spatial point processes are the Binomial and Poisson point processes as described below, see details in [123].

Definition 5.1 (Finite Binomial Point Process (FBPP) [113, 123]). *A Finite Binomial Point Process is defined by considering a fixed number of n points at random locations in a bounded region $W \subset \mathbb{R}^2$. Define by $X_1, \dots, X_n$ the i.i.d. random locations with the intensity of the number of points in a small region around any location $\mathbf{x}$ denoted as $\lambda(\mathbf{x})$. This produces a probability density of each X_i given by*

$$f_X(x) = \begin{cases} \frac{1}{\lambda(W)}, & \text{if } x \in W, \\ 0, & \text{otherwise,} \end{cases} \quad (5.1)$$

where $\lambda(W)$ denotes the area of W . Each random point X_i is uniformly distributed in W so that for a bounded set $B \in \mathbb{R}^2$ has the distribution

$$\Pr(X_i \in B) = \int_B f_X(x) dx = \frac{\lambda(B \cap W)}{\lambda(W)}. \quad (5.2)$$

Under this model one may define the Binomial point process by the random variables for counts and vacancies for sets B given by

$$\begin{aligned} N(B) &= \sum_{i=1}^n \mathbb{I}[X_i \in B] \\ V(B) &= \min_{i=1}^n \mathbb{I}[X_i \notin B]. \end{aligned} \quad (5.3)$$

Then it can be shown that $N(B) \sim \text{Binomial}(n, p)$ where

$$p = \frac{\lambda(B \cap W)}{\lambda(W)}. \quad (5.4)$$

The following is also known about the Binomial spatial point process for disjoint sets B_i and B_j that result from the non-independence property for $N(B_i)$ and $N(B_j)$.

Remark 5.1. *It can be observed that the counting variable $N(B)$ for different subsets B are not independent. In general the joint distribution for two subsets B_1 and B_2 given n trials satisfies $(N(B_1), N(B_2)) \sim \text{Multinomial}(n, p_1, p_2)$ with*

$$p_i = \frac{\lambda(B_i \cap W)}{\lambda(W)}. \quad (5.5)$$

In this thesis we will consider a general case involving an inhomogeneous version of the finite domain spatial binomial point process (inhomogeneous FBPP). The distribution of the points has a density $f_X(x)$ given by the decaying power law, relative to the center of W which is specified to be a disc in our applications. Therefore it will have a form given according to

$$f_X(x) = \begin{cases} \mathbf{x}^{-\nu} \mathbf{1}(\mathbf{x}), & \text{if } x \in W \\ 0, & \text{otherwise,} \end{cases} \quad (5.6)$$

with reference to the center of W .

The second point process to be considered will be on an infinite domain and will be considered to be an Infinite Poisson point process defined according to Definition 5.2.

Definition 5.2 (Infinite Poisson Point Process (IPPP) [113]). *Consider a locally compact metric space $W \subseteq \mathbb{R}^2$ and measure Λ on W which is finite on every compact set and contains no atoms. Then the spatial Poisson point process on W with intensity measure Λ is a point process on W such that*

- *for every compact set $B \subset W$, the count $N(B)$ is distributed according to a Poisson distribution with $\Lambda(B)$ mean; and*
- *if $B_1, \dots, B_m$ are disjoint compact sets, then $N(B_1), \dots, N(B_m)$ are independent.*

In the examples considered in this thesis we will utilise one of two possibilities:

- Homogeneous IPPP (HIPPP), when λ is constant.
- Non-homogeneous IPPP (NIPPP), when λ is not constant. Without loss of generality, we also consider a power law $\lambda(\mathbf{x}) = \mathbf{x}^{-\nu}$ with $\nu > 0$.

The difference between the FBPP and the IPPP is that the number of points n within set V , is a *known constant* under FBPP, and is a *random unknown* under IPPP. Fig. 5.1 presents realisations from homogenous BPP, inhomogeneous BPP, homogenous PPP and inhomogeneous PPP, respectively.

Figure 5.1: Realisations from Homogenous BPP deployment (top left), Non-homogenous BPP deployment (bottom left), Homogenous PPP deployment (top right) and Non-homogenous PPP deployment (bottom right) .

5.1.1 Wireless Sensor Network Operation Model

We now present the system model for the WSN.

- (i) The source is present ($\mathcal{H}_1$) or absent ($\mathcal{H}_0$). Under $\mathcal{H}_1$, the source transmits at a constant power P_0 , and under $\mathcal{H}_0$, the source does not transmit any message ($P_0 = 0$).
- (ii) The location of the source (if present) is assumed to be known at $\mathbf{x}_s = [x_0, y_0]$. We assume, without loss of generality, that it is located at the center of circle of radius R .
- (iii) Consider a WSN consisting of sensors with locations following either a FBPP deployment (Definition 5.1) or an IPPP deployment (Definition 5.2) in a 2 dimensional region.
 - (a) For the FBPP deployment, N_B sensors are deployed in a circle with radius R .

- (b) For the IPPP deployment, an unknown random number of sensors are deployed in a circle with radius R . Note that for the case of IPPP, the number of points N_P is not fixed.

The spatial density of the sensors is given by $\lambda(\mathbf{x}) = \mathbf{x}^{-\nu}$, $\nu \neq 0$.

- (iv) The unknown random location of the k -th sensor ($k = \{1, \dots, N\}$) is $\mathbf{X}_k = [X_k, Y_k]$.
- (v) The amount of energy the k -th sensor measures is inversely proportional to the Euclidean distance between the source and the sensor and is given by $\sqrt{P_0} R_k^{-\alpha/2}$. The random variable R_k represents the random distance between the k -th sensor and the source. This distance is defined as the minimum radius of the ball with center $\mathbf{x}_s$, that contains at least k points in the ball, i.e., $R_k = \inf\{r : \{R_{(1)}, R_{(2)}, \dots, R_{(k)}\} \in B_{\mathbf{x}_s}(r)\}$. $B_{\mathbf{x}_s}(r)$ is the ball with radius r and center at $\mathbf{x}_s$.
- (vi) Each sensor transmits its observation over a perfect channel to the GW via a shared medium. The observed signal at the GW in the l -th time slot ($l = \{1, \dots, L\}$) is a linear combination of all the signals given by:

$$\begin{cases} \mathcal{H}_0 : Y_l = W_l \\ \mathcal{H}_1 : Y_l = \sum_{k=1}^{N_i} \sqrt{P_0} R_k^{-\alpha/2} + W_l, \end{cases}$$

where W_l is the i.i.d additive Gaussian noise $\mathcal{N}(0, \sigma_{W_l}^2)$. The parameter α is the path-loss coefficient.

The optimal decision rule for the event detection is presented, followed by the various components required in order to evaluate the optimal decision rule.

5.1.2 Optimal Event Detection Decision Rule

The optimal decision rule is a threshold test based on the likelihood ratio [116]. We consider a frame-by-frame detection, where the length of each frame is L . The decision rule is then given by:

$$\Lambda(Y_{1:L}) \triangleq \frac{p(Y_{1:L}|\mathbf{x}_s, \mathcal{H}_0)}{p(Y_{1:L}|\mathbf{x}_s, \mathcal{H}_1)} \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\gtrless}} \gamma, \quad (5.7)$$

where the threshold γ can be set to assure a fixed system false-alarm rate under the Neyman-Pearson approach or can be chosen to minimize the overall probability of detection error under the Bayesian approach [124]. We can decompose the full marginals under each hypothesis, $p(Y_{1:L}|\mathbf{x}_s, \mathcal{H}_k)$, $k = 0, 1$, as

$$p(Y_{1:L}|\mathbf{x}_s, \mathcal{H}_k) = \prod_{l=1}^L p(Y_l|\mathbf{x}_s, \mathcal{H}_k).$$

This decomposition is useful as it allows us to work on a lower dimensional space, resulting in efficiency gains for the algorithm we develop and requiring no memory storage for data.

5.2 Event Detection Algorithm under Random Point Process Sensor Deployment

The optimal decision rule in (5.7) involves calculating the marginal likelihood under each hypothesis, $p(Y_{1:L}|\mathbf{x}_s, \mathcal{H}_k)$, $k = 0, 1$. The marginal likelihood under $\mathcal{H}_0$ can be easily calculated as it follows a Normal distribution. The marginal likelihood under the alternative hypothesis, $p(Y_l|\mathbf{x}_s, \mathcal{H}_1)$, is not attainable in closed form because it involves solving the $(N + 1)$ -fold convolution as we will show in this section (see (5.8)). We will therefore develop a novel approximation of the marginal likelihood under the alternative hypothesis, based on an Askey-orthogonal polynomial expansion. In particular, we will derive the Gram-Charlier, Gamma-Laguerre and Beta-Jacobi type series expansions.

We begin with obtaining the distribution of the distance between the k -th sensor and the source location, denoted by $f_{R_k}(r|\mathbf{x}_s, \mathcal{H}_1)$. We derive this density for both BPP and IPPP deployments. To achieve that we extend the earlier work of [68] which derived the distance distribution for the homogeneous case, to the inhomogeneous case.

Theorem 5.1. *The density of the Euclidean distance between the k -th sensor and the source, R_k , is given by:*

(i) *BPP deployment:*

$$f_{R_k}(r|\mathbf{x}_s, \mathcal{H}_1) = \frac{(2 - \nu) \Gamma(k + \frac{1-\nu}{2-\nu}) \Gamma(N_B + 1)}{R \Gamma(k) \Gamma(N_B + \frac{1-\nu}{2-\nu} + 1)} \beta \left(\left(\frac{r}{R} \right)^{2-\nu}; k + \frac{1-\nu}{2-\nu}, N_B - k + 1 \right).$$

(ii) *IPPP deployment:*

$$f_{R_k}(r|\mathbf{x}_s, n = N_P, \mathcal{H}_1) = \frac{(2\pi)^k}{\Gamma(k)(2-\nu)^{k-1}} r^{(2-\nu)k-1} \exp\left(-\frac{2\pi r^{2-\nu}}{2-\nu}\right),$$

where $\beta(x, \alpha, \beta) := \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1}$ is the β distribution and $\Gamma(n) := (n-1)!$ is the Gamma function. For both cases, the support of R_k is $Z_k \in R^+$.

Proof. See Appendix A.1. □

Corollary 5.1. When $N \rightarrow +\infty$ and $R \rightarrow +\infty$, the non-homogenous BPP converges to an IPPP. Since $p = (r/R)^{2-\nu}$, then $R = rp^{-\frac{1}{2-\nu}}$. Also, $N = \int_W \lambda(r)dr = \frac{2\pi}{2-\nu} R^{2-\nu}$. We have $p = (r/R)^{2-\nu} = \frac{2\pi r^{2-\nu}}{N(2-\nu)}$. The density f_{R_k} is given by:

$$\begin{aligned} f_{R_k}(r|\mathbf{x}_s, \mathcal{H}_1) &= \lim_{N \rightarrow +\infty} \frac{-\nu + 2(1-p)^{N-k} p^k \Gamma(N+1)}{r \Gamma(N-k+1) \Gamma(k)} \\ &= \frac{2-\nu}{r} \lim_{N \rightarrow +\infty} \frac{\left(1 - \frac{2\pi r^{2-\nu}}{N(2-\nu)}\right)^{N-k} \left(\frac{2\pi r^{2-\nu}}{N(2-\nu)}\right)^k \Gamma(N+1)}{\Gamma(N-k+1) \Gamma(k)} \\ &= \frac{2-\nu}{\Gamma(k)r} \left(\frac{2\pi r^{2-\nu}}{2-\nu}\right)^k \exp\left(-\frac{2\pi r^{2-\nu}}{2-\nu}\right) \lim_{N \rightarrow +\infty} \frac{\Gamma(N+1)}{N^k \Gamma(N-k+1)} \\ &= \frac{(2\pi)^k}{\Gamma(k)(2-\nu)^{k-1}} r^{(2-\nu)k-1} \exp\left(-\frac{2\pi r^{2-\nu}}{2-\nu}\right) \lim_{N \rightarrow +\infty} \frac{N!}{N^k (N-k)!} \\ &= \frac{(2\pi)^k}{\Gamma(k)(2-\nu)^{k-1}} r^{(2-\nu)k-1} \exp\left(-\frac{2\pi r^{2-\nu}}{2-\nu}\right) \lim_{N \rightarrow +\infty} \frac{\prod_{i=0}^{k-1} (N-i)}{N^k} \\ &= \frac{(2\pi)^k}{\Gamma(k)(2-\nu)^{k-1}} r^{(2-\nu)k-1} \exp\left(-\frac{2\pi r^{2-\nu}}{2-\nu}\right) \mathcal{O}\left(\frac{N^k}{N^k}\right) \\ &= \frac{(2\pi)^k}{\Gamma(k)(2-\nu)^{k-1}} r^{(2-\nu)k-1} \exp\left(-\frac{2\pi r^{2-\nu}}{2-\nu}\right). \end{aligned}$$

Next, based on the result in Theorem 5.1, we derive the density distribution of each of the elements in the observation under the alternative hypothesis Y_l . This involves the non-linear transformation of the random distance, namely $f_{Z_k}(r^{-\alpha/2}|\mathbf{x}_s, \mathcal{H}_1)$.

Lemma 5.1. The density $f_{Z_k}(z|\mathbf{x}_s, \mathcal{H}_1) = f_{R_k}(r^{-\alpha/2}|\mathbf{x}_s, \mathcal{H}_1)$ is given by:

(i) *BPP deployment:*

$$\begin{aligned} f_{Z_k}(z|\mathbf{x}_s, \mathcal{H}_1) &= \frac{2(2-\nu)}{\alpha R} \frac{\Gamma(k + \frac{1-\nu}{2-\nu}) \Gamma(N_B + 1)}{\Gamma(k) \Gamma(N_B + \frac{1-\nu}{2-\nu} + 1)} \\ &\quad \times \beta\left(\left(\frac{z^{-2/\alpha}}{R}\right)^{2-\nu}; k + \frac{1-\nu}{2-\nu}, N_B - k + 1\right) z^{-2/\alpha-1}. \end{aligned}$$

(ii) *IPPP deployment:*

$$f_{Z_k}(z|\mathbf{x}_s, n = N_P, \mathcal{H}_1) = \frac{(2\pi)^k}{\Gamma(k)(2-\nu)^{k-1}} \left(\frac{2}{\alpha}\right) z^{-2/\alpha((2-\nu)k)-1} \exp\left(-\frac{2\pi}{2-\nu} z^{-2/\alpha(2-\nu)}\right).$$

For both cases, the support of Z_k is $Z_k \in (R^{-\alpha/2}, \infty)$.

Proof. See Appendix A.2. □

Remark 5.2. In theory, Z_k is from $(R^{-\alpha/2}, \infty)$. In simulation, we let $r \geq R_c$, where R_c is the radius of a critical region which we will show in Section 5.5.3. So Z_k is bounded within $(R^{-\alpha/2}, R_c^{-\alpha/2})$.

Now that we have derived the density and distribution of each of the elements in Y_l , we need to derive the density of the term $\sum_{k=1}^{N_i} \sqrt{P_0} Z_k$. We express this random sum of Y as an N -fold convolution of $Z_k, k \in \{1, \dots, N\}$, given by

$$f_Y(y) = *_{i=1}^{N_i} f_{Z_i}(y) = \int_{-\infty}^{\infty} f_{Z_{N_i-1}}(y-w) f_{Z_{N_i}}(w) dw, \quad (5.8)$$

where $*$ represents the convolution operation. Each of these convolution integrals is intractable and cannot be solved analytically in closed form. To approximate the marginal likelihood Y_l we will utilise a series expansion approach presented in the next section.

5.3 Probability Density Approximation via Series Expansion Methods

In order to evaluate the marginal likelihood in (5.8), we derive novel approximations for the marginal likelihood. We develop three different series expansion methods for representing the marginal likelihood using orthogonal basis functions [73, 125]. We will show how these expansions are applicable under different scenarios. The series expansion we utilise are based on a kernel density multiplied by polynomials, known as Askey polynomials [126]. Typical Kernel densities include Gaussian density basis, Gamma density basis and Beta density basis. The respective Askey polynomials [126] are Hermite polynomials, Laguerre polynomials and

Jacobi polynomials. These series expansions for the scalar case can be generically expressed as follows:

$$f(y) = g(y) \left(1 + \sum_{j=1}^{\infty} d_j H_j(y) \right), \quad (5.9)$$

where $g(y)$ is the kernel, d_j is the j -th weight and $H_j(y)$ is the j -th order basis function. All of these series expansion methods use the basic properties of orthogonality between density functions and polynomials. This property guarantees the integration of density to be equal to one [73, 125]. Each of these series expansion has different properties and different supports. An important aspect of these expansions is that they do not ensure positivity of the density at all points (for example, it can be negative for particular choices of Skew and Kurtosis). It is therefore important to characterize these values that produce the "envelope" for the density approximation in which it will remain positive. This characterization can be carried out by finding the appropriate regions in the Skew-Kurtosis plane (S-K plane) which generate positive support [73, 125].

We will derive three series expansion methods, which will be used for different practical scenarios (e.g., different system parameters). The first two are the Gram-Charlier and Gamma-Laguerre series expansions. We then develop a new series expansion which we term the Beta-Jacobi series expansion.

5.3.1 Gram-Charlier Series Expansion

The Gram-Charlier series expansion utilises a Gaussian kernel, $g(y)$, and Hermite polynomials, $H_s(x)$, as basis functions. These polynomials are defined in terms of the derivatives of the normal density, $g(y)$ as follows:

$$\frac{d^s g(y)}{d^s y} = (-1)^s H_s(y) g(y). \quad (5.10)$$

The Gram-Charlier series expansion is given by:

$$f_Y(y) = \frac{1}{\sqrt{2\pi\kappa_2}} \exp\left(-\frac{(y - \kappa_1)^2}{2\kappa_2^2}\right) \sum_{r=1}^{\infty} \frac{\kappa_r}{r!} \frac{-d^r(g(y))}{d^r y}, \quad (5.11)$$

where $g(y)$ is the normal density and κ_r is the r -th cumulant of Y .

If we include only the first two correction terms to the normal distribution, we obtain the *Gram-Charlier A* series presented next.

Lemma 5.2 (Gram-Charlier A Series Expansion:). *The fourth order approximation of a probability distribution, $f_Y(y)$, via the Gram-Charlier A series is given by*

$$f_Y(y) \approx \frac{1}{\sqrt{2\pi\kappa_2}} \exp\left(-\frac{(y-\kappa_1)^2}{2\kappa_2^2}\right) \left(1 + \frac{\kappa_3}{6\kappa_2^3} H_3\left(\frac{y-\kappa_1}{\kappa_2}\right) + \frac{\kappa_4}{24\kappa_2^4} H_4\left(\frac{y-\kappa_1}{\kappa_2}\right)\right), \quad (5.12)$$

where $H_3(y) = y^3 - 3y$ and $H_4(y) = y^4 - 6y^2 + 3$ are the Hermite polynomials, and $\kappa_1, \kappa_2, \kappa_3, \kappa_4$ are the first, second, third and fourth cumulants of Y .

As mentioned before, it is important to characterise the regions in the S-K plane which yield positive support of the density. This is presented in the following Lemma.

Lemma 5.3 (Positive density conditions [73]:). *The Gram-Charlier A series expansion yields positive values for the density $f_Y(y)$ only if:*

$$\begin{cases} s(\tilde{y}) &= -24 \frac{H_3(\tilde{y})}{d(\tilde{y})}, \\ k(\tilde{y}) &= 72 \frac{H_2(\tilde{y})}{d(\tilde{y})}, \end{cases}$$

where $\tilde{Y} = \frac{Y-\kappa_1}{\kappa_2}$ and $d(\tilde{y}) = 4H_3^2(\tilde{y}) - 3H_2(\tilde{y})H_4(\tilde{y})$.

This region characterizes the positive density regions for the Gram-Charlier series in terms of the Skew and Kurtosis properties of the approximated distribution. The support of the Gram-Charlier expansion is $y \in \mathbb{R}$.

5.3.2 Gamma-Laguerre Series Expansion

The Gamma-Laguerre series expansion approximates a probability distribution, $f_Y(y)$, by utilising the orthogonality between the Gamma density kernel and the Laguerre polynomials in order to obtain an efficient series expansion [125]. In contrast to the Gram-Charlier series expansion where the Hermite polynomials have support on the entire real line, the Laguerre polynomials only have support on the positive real line $y \in \mathbb{R}^+$.

Instead of directly working with y , we first rescale it to a Random Variable (R.V.) $\tilde{y}$ by $\tilde{y} = by$, where $b = \frac{\mathbb{E}[y]}{\text{Var}[y]}$ and set $a = \frac{\mathbb{E}[y]^2}{\text{Var}[y]}$. Denoting the density of $\tilde{y}$ as $f_{\tilde{y}}$, we express $f_{\tilde{y}}$ as follows:

$$f_{\tilde{y}}(\tilde{y}) = g(\tilde{y}; a) \sum_{n=1}^{\infty} A_n L_n^{(a)}(\tilde{y}),$$

where the kernel is the Gamma density, i.e., $g(\tilde{y}; a) = \frac{\tilde{y}^{a-1} \exp(-\tilde{y})}{\Gamma(a)}$, with shape = a and scale = 1, and the orthonormal polynomial basis (with respect to this kernel) is given by the Laguerre polynomials (in contrast to Hermite polynomials in the Gaussian case of the Gram-Charlier expansion), defined as

$$L_n^{(a)}(x) = (-1)^n x^{1-a} \exp(-x) \frac{d^n}{dx^n} (x^{n+a-1} \exp(-x)).$$

The first five orthonormal polynomial basis are given by:

$$L_0^{(a)}(x) = 1$$

$$L_1^{(a)}(x) = x - a$$

$$L_2^{(a)}(x) = x^2 - 2(a+1)x + (a+1)a$$

$$L_3^{(a)}(x) = x^3 - 3(a+2)x^2 + 3(a+2)(a+1)x - (a+2)(a+1)a$$

$$L_4^{(a)}(x) = x^4 - 4(a+3)x^3 + 6(a+3)(a+2)x^2 - 4(a+3)(a+2)(a+1)x + (a+3)(a+2)(a+1)a$$

and $A_0 = 1, A_1 = 0, A_2 = 0, A_3 = \frac{\Gamma(a)}{3!\Gamma(a+3)}(\mu_3 - 2a)$ and $A_4 = \frac{\Gamma(a)}{4!\Gamma(a+4)}(\mu_4 - 12\mu_3 - 3a^2 + 18a)$.
 $\mu_n = \mathbb{E}[(X - \mathbb{E}[X])^n], n = 3, 4.$

Next we characterise the S-K region of the Gamma-Laguerre series expansion in which it yields positive support. These results are based on those derived in [125].

Lemma 5.4 (Positive density conditions:). *The Gamma-Laguerre series expansion yields positive values for the density $f_X(x)$ if:*

$$\begin{cases} k(x) = (\frac{B'_1 B_3}{B_1} - B'_3)(B'_2 - \frac{B'_1 B_2}{B_1})^{-1} \\ s(x) = -\frac{1}{B_1}(\mu_4(x)B_2 + B_3) \end{cases}, \text{ for } x \in [0, +\infty)$$

where $B_1, B_2, B_3, B'_1, B'_2$ and B'_3 are defined in [125] and are given by:

$$\begin{aligned}
 B_1(x) &= \frac{x^{a-1} \exp(-x)}{\Gamma(a)} \left(\frac{\Gamma(a)}{3!\Gamma(a+3)} L_3^{(a)}(x) - 12 \frac{\Gamma(a)}{4!\Gamma(a+4)} L_4^{(a)}(x) \right), \\
 B_2(x) &= \frac{x^{a-1} \exp(-x)}{\Gamma(a)} \frac{\Gamma(a)}{4!\Gamma(a+4)} L_4^{(a)}(x), \\
 B_3(x) &= \frac{x^{a-1} \exp(-x)}{\Gamma(a)} \left(1 - 2a \frac{\Gamma(a)}{3!\Gamma(a+3)} L_3^{(a)}(x) + (-3a^2 + 18a) \frac{\Gamma(a)}{4!\Gamma(a+4)} L_4^{(a)}(x) \right), \\
 B'_1(x) &= ((a-1)x^{-1} - 1) B_1(x) + \frac{x^{a-1} \exp(-x)}{\Gamma(a)} \left(\frac{\Gamma(a)}{3!\Gamma(a+3)} \frac{dL_3^{(a)}}{dx}(x) - 12 \frac{\Gamma(a)}{4!\Gamma(a+4)} \frac{dL_4^{(a)}}{dx}(x) \right), \\
 B'_2(x) &= ((a-1)x^{-1} - 1) B_2(x) + \frac{x^{a-1} \exp(-x)}{\Gamma(a)} \left(\frac{\Gamma(a)}{4!\Gamma(a+4)} \frac{dL_4^{(a)}}{dx}(x) \right), \\
 B'_3(x) &= ((a-1)x^{-1} - 1) B_3(x) + \frac{x^{a-1} \exp(-x)}{\Gamma(a)} \\
 &\quad \times \left(-2a \frac{\Gamma(a)}{3!\Gamma(a+3)} \frac{dL_3^{(a)}}{dx}(x) + (-3a^2 + 18a) \frac{\Gamma(a)}{4!\Gamma(a+4)} \frac{dL_4^{(a)}}{dx}(x) \right),
 \end{aligned}$$

$$\begin{aligned}
 \frac{dL_3^{(a)}}{dx}(x) &= 3x^2 - 6(a+2)x + 3(a+2)(a+1), \\
 \frac{dL_4^{(a)}}{dx}(x) &= 4x^3 - 12(a+3)x^2 + 12(a+3)(a+2)x - 4(a+3)(a+2)(a+1).
 \end{aligned}$$

5.3.3 Beta-Jacobi Series Expansion

In this section we develop a novel series expansion that is based on the Beta kernel. This new expansion is relevant for cases where Y has a bounded support $[a, b]$. To achieve this, we construct the series based on a Beta kernel (instead of Gamma or Normal kernels as before) and the Jacobi polynomials. It is important to note that the Jacobi polynomials are only orthogonal on $[-1, 1]$. Hence, we need to transform Y so that it also has support $[-1, 1]$. This is achieved via the transformation

$$X = \frac{2}{b-a} \left(Y - \frac{a+b}{2} \right). \quad (5.13)$$

We now present our novel Beta-Jacobi density series expansion, see discussions in [127].

Theorem 5.2 (Beta-Jacobi density series expansion). *The Beta-Jacobi series expansion is given by:*

$$f_X(x) = \frac{(x+1)^{\theta-1} (1-x)^{\eta-1}}{B(\theta, \eta) 2^{\theta+\eta-1}} \sum_{i=0}^d a_i P_i^{(\eta-1, \theta-1)}(x),$$

where the coefficients, a_i , and the Jacobi polynomials, $P_i^{(\eta-1, \theta-1)}(x)$, are given by:

$$a_i = \sum_{j=0}^i \mathbb{E}[X^j] \frac{B(\theta, \eta) (2i + \theta + \eta - 1) i!}{\Gamma(i + \theta)} \sum_{m=j}^i \frac{\Gamma(\eta + \theta + i + m - 1)}{\Gamma(i - m + 1) \Gamma(\eta + m) m! 2^m} \binom{m}{j} (-1)^{m-j}$$

$$P_i^{(\eta-1, \theta-1)}(x) = \frac{\Gamma(\eta + i)}{\Gamma(\eta + \theta + i - 1)} \sum_{m=0}^i \frac{\Gamma(\eta + \theta + i + m - 1)}{\Gamma(i - m + 1) \Gamma(\eta + m + 1) m!} \left(\frac{x-1}{2}\right)^m.$$

Proof. See Appendix A.3. □

The distribution of Y is obtained from the distribution of X via the transformation

$$f_Y(y) = \frac{2}{b+a} f_X\left(\frac{2(y+a)}{b+a} - 1\right), \quad (5.14)$$

where $f_X(x)$ is given in Theorem 5.2. The values of θ, η need to be chosen in order to find a good approximation of X . We find approximate values of θ, η through K-S curve as mentioned below.

Next we find the Skew-Kurtosis conditions to guarantee positive density:

Theorem 5.3 (Positive density conditions:). *The Beta-Jacobi series expansion yields positive values for the density $f_X(x)$ if:*

$$\begin{cases} k(x) &= (\frac{B'_1 B_3}{B_1} - B'_3)(B'_2 - \frac{B'_1 B_2}{B_1})^{-1} \\ s(x) &= -\frac{1}{B_1}(\mu_4(x) B_2 + B_3) \end{cases}, \text{ for } x \in [-1, +1)$$

where $B_1, B_2, B_3, B'_1, B'_2, B'_3$ are defined in Appendix A.4.

Proof. See Appendix A.4 □

Remark 5.3. *To use the three series expansions above, we have to calculate the first four cumulants and moments of the model under IBPP and IPPP, which will be presented this in the next section.*

5.4 Calculation of The Moments

As mentioned earlier, the series expansions we developed require the cumulants of Y under the alternative hypothesis. To obtain the cumulants we need to calculate the Moment Generating Function (MGF) of the observation $Y_l|\mathcal{H}_1$, given by

$$M_{Y_l|\mathcal{H}_1}(t) = M_{Z_1}(t) M_{Z_2}(t) \cdots M_{Z_N}(t) M_{W_l}(t).$$

Calculating the MGF of each of the elements, Z_k , as presented in Lemma 5.1 for BPP and IPPP, involves the following:

(i) BPP deployment:

$$M_{Z_k}(t) = \mathbb{E}_{f_{Z_k}}[\exp(tz)] = \int w(k) \beta \left(\left(\frac{z^{-2/\alpha}}{R} \right)^{2-\nu}; k + \frac{-\nu}{1-\nu}, N_i - k + 1 \right) z^{-2/\alpha-1} \exp(tz) dz,$$

where

$$w(k) = \frac{2(2-\nu)}{\alpha R} \frac{\Gamma(k + \frac{1-\nu}{2-\nu}) \Gamma(N_B + 1)}{\Gamma(k) \Gamma(N_B + \frac{1-\nu}{2-\nu} + 1)}.$$

(ii) IPPP deployment:

$$M_{Z_k}(t) = \mathbb{E}_{f(Z_k)}[\exp(tZ)] = \int \frac{(2\pi)^k}{\Gamma(k)(2-\nu)^{k-1}} \left(\frac{2}{\alpha} \right) z^{-2/\alpha((2-\nu)k)-1} \exp\left(-\frac{2\pi}{2-\nu} z^{-2/\alpha(2-\nu)}\right) dz.$$

As it is difficult to solve both integrals directly, an equivalent solution is obtained instead by calculating the m -th moment for Z_k and then deriving the MGF based on the moments.

Theorem 5.4. *The m -th moment of Z_k is given by:*

(i) BPP deployment:

$$\mathbb{E}[Z_k^m] = \begin{cases} R^{-m\alpha/2} \frac{\Gamma(N_B+1) \Gamma(k - \frac{m\alpha}{2(2-\nu)})}{\Gamma(k) \Gamma(N_B - \frac{m\alpha}{2(2-\nu)} + 1)}, & k - \frac{m\alpha}{2(2-\nu)} \notin \mathbb{Z}_{\leq 0} \\ \infty, & \text{otherwise} \end{cases} \quad (5.15)$$

(ii) IPPP deployment:

$$\mathbb{E}[Z_k^m] = \begin{cases} \left(\frac{\nu+2}{2\pi} \right)^{-\frac{\alpha m}{2(\nu+2)}} \frac{\Gamma(k - \frac{\alpha m}{2(\nu+2)})}{\Gamma(k)}, & k - \frac{\alpha m}{2(\nu+2)} \notin \mathbb{Z}_{\leq 0} \\ \infty, & \text{otherwise} \end{cases}$$

Proof. See Appendix A.5 □

Based on the moments, we can calculate the cumulants and moments.

Lemma 5.5. *The first four cumulants of $Y_l|\mathbf{x}_s, \mathcal{H}_1, \kappa_i, i = 1, 2, 3, 4$ are given by:*

$$\begin{aligned}\kappa_1 &= \sum_{k=1}^{N_B} \sqrt{P_0} \mathbb{E}[Z_k], \\ \kappa_2 &= \sum_{k=1}^{N_B} P_0 \left(\mathbb{E}[Z_k^2] - (\mathbb{E}[Z_k])^2 \right) + \sigma_W^2, \\ \kappa_3 &= \sum_{k=1}^{N_B} \sqrt{P_0}^3 \left(\mathbb{E}[Z_k^3] - 3\mathbb{E}[Z_k^2] \mathbb{E}[Z_k] + 2(\mathbb{E}[Z_k])^3 \right), \\ \kappa_4 &= \sum_{k=1}^{N_B} P_0^2 \left(\mathbb{E}[Z_k^4] - 4\mathbb{E}[Z_k^3] \mathbb{E}[Z_k] - 3(\mathbb{E}[Z_k^2])^2 + 12\mathbb{E}[Z_k^2] (\mathbb{E}[Z_k])^2 - 6(\mathbb{E}[Z_k])^4 \right).\end{aligned}$$

The Moments can be expressed as polynomials of cumulants:

$$\begin{aligned}\mu_1 &= \kappa_1, \\ \mu_2 &= \kappa_2 + \kappa_1^2, \\ \mu_3 &= \kappa_3 + 3\kappa_2\kappa_1 + \kappa_1^3, \\ \mu_4 &= \kappa_4 + 4\kappa_3\kappa_1 + 3\kappa_2^2 + 6\kappa_2\kappa_1^2 + \kappa_1^4.\end{aligned}$$

Proof. See Appendix A.6. □

Now that we have derived the four cumulants and moments, we use the series expansion methods in Section 5.3 to approximate $p_{Y_l}(y|\mathbf{x}_s, \mathcal{H}_1)$ and derive the LRT in (5.7). Finally, the Event Detection algorithm under BPP and IPPP is presented in Algorithm 5.1.

5.5 Simulation Results

We present the performance of the proposed algorithms via Monte Carlo simulations in term of the following metrics:

- (i) **Moment calculation accuracy:** the series expansions we develop are based on fourth order moments and cumulants. Hence it is important to verify the accuracy of the results which are based on Theorem 5.1, Lemma 5.1, Theorem 5.4 and Lemma 5.5.

Algorithm 5.1 Event Detection in Sensor Networks with Random Deployment

Input: $Y_l, \gamma, N_B, R, \nu, \alpha, \sigma_w$

Output: Binary decision $(\mathcal{H}_1, \mathcal{H}_0)$

- (i) Calculate the first four moments according to Theorem 5.4.
 - (ii) Calculate κ_i and μ_i according the Lemma 5.5.
 - (iii) Perform S-K region analysis to assess the appropriateness of each of the series expansions according to Lemma 5.3, Lemma 5.4 and Theorem 5.3.
 - (iv) Choose the series expansion for which the S-K point is inside the K-S region.
 - (v) Evaluate the series expansion chosen in Section 5.3 to find $\widehat{p}_{Y_l}(y|\mathbf{x}_s, \mathcal{H}_1)$ according to (5.9).
 - (vi) Calculate $\Lambda(Y_l)$ via (5.7) and compare to the threshold γ .
-

- (ii) **Positivity regions of the estimated density:** we characterise the regions at which the density has positive support, based on Lemma 5.3, Lemma 5.4 and Theorem 5.3. We then show the implications of not choosing the correct series expansion.
- (iii) **Detection performance:** we present the detection and false alarm probability via Receiver Operating Characteristics (ROC) curves under different scenarios.

The simulations setting is as follows: the results are obtained from 50,000 realizations for a given parameter set of $N, \sigma_w, \nu, P_0, R, \alpha$. The additive noise is assumed to be i.i.d Gaussian distributed at each sensor.

5.5.1 Performance of Event Detection Algorithm using Gram-Chalier Series Expansion

We first present the detection performance of our algorithm via Monte Carlo simulations using Gram Chalier series expansion. The simulations setting are as follows: the additive noise is assumed i.i.d Gaussian distributed at each sensor. The results are obtained from 50,000 realizations for a given parameter set of $N, \sigma_w, P_0, R, \alpha$.

First, in Fig. 5.2 we present an example of the Gram-Chalier series expansion approximation of $p(Y(s)|\mathbf{X}_s, \mathcal{H}_1)$. The result indicates that the Monte Carlo density estimation and

the Gram-Chalier series expansion are in perfect agreement. This shows the effectiveness and accuracy of our approach. We note that similar results were obtained for a large number of parameters.

Figure 5.2: Probability density estimation of $p(Y(s) | \mathbf{X}_s, \mathcal{H}_1)$ using Monte Carlo simulation and Gram-Chalier series expansion ($R = 100, N = 50, \alpha = 2, \sigma_w = 0.01, P_0 = 1$).

We now present the detection performance of the algorithms via Receiver Operating characteristics (ROC) for different configurations: Gaussian Noise variance $\sigma_w = \{0.005, 0.01, 0.015, 0.02, 0.025, 0.03\}$, source power $P_0 = \{1, 2, 3, 4, 5\}$, field radius $R = \{50, 60, 70, 80, 90, 100\}$ and propagation loss coefficient $\alpha = \{2, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6\}$.

Fig. 5.3 presents ROC results for various values of the noise variance σ_w . This result clearly shows the effect the additive noise has on the ROC performance.

Fig. 5.4 presents ROC performance as a function of the transmitted power P_0 . The result shows that with the increase of P_0 , the performance of the algorithm improves. This is expected as high P_0 at the source will make it easier for the algorithm to distinguish between the two hypotheses.

Fig. 5.5 presents ROC results as the radius R of the deployed field varies. As expected, when the other parameters are fixed, the performance of algorithm improves with decreasing R . This is as R increases, the signals travel larger distances and as a result decay more, which makes it harder for GW to make a correct decision.

Figure 5.3: ROC performance under different configurations of σ_w $\sigma_w \in \{0.005, 0.01, 0.015, 0.02, 0.025, 0.03\}$, ($R = 100, \alpha = 2, N = 50, P_0 = 1$)

Figure 5.4: ROC performance under different configurations of P_0 , $P_0 \in \{1, 2, 3, 4, 5\}$, ($R = 100, \alpha = 2, \sigma_w = 0.01, N = 50$)

Fig. 5.6 presents ROC results as the propagation loss coefficient α varies. When the other parameters are fixed, the performance of algorithm improves with decreasingly α . This is because the smaller α is, the stronger the signal the sensor observes, which makes it easier for GW to make a correct decision.

Figure 5.5: ROC performance under different configurations of radius $R \in \{50, 60, 70, 80, 90, 100\}$, ($N = 50, \alpha = 2, \sigma_w = 0.01, P_0 = 1$)

Figure 5.6: ROC performance under different configurations of propagation loss coefficient $\alpha \in \{2, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6\}$, ($R = 100, N = 50, \sigma_w = 0.01, P_0 = 1$)

5.5.2 Moments Calculation

In Figs. 5.7-5.8, we present comparison of theoretical moments (Theorem 5.4) and Monte Carlo simulations for BPP and PPP, respectively. For both cases we consider both homogeneous and non-homogenous deployments. For both BPP and PPP homogeneous deployments,

the results show perfect agreement. For non-homogenous PPP, the result slightly disagrees. This is due to the generation of non-homogenous PPP. Specifically,

Figure 5.7: Comparison of the theoretical results of the first four moments per Theorem 5.4 and the corresponding Monte Carlo simulation results under BPP type deployment: (a) Homogeneous BPP deployments. The parameters used: $R = 100, \alpha = 0.5, N = 10, \nu = \{0, 0.5\}$; (b) Non-homogeneous BPP deployments. The parameters used: $R = 100, \alpha = 0.5, \nu = \{0, 0.5\}$.

Remark 5.4. Given $\lambda(x) = x^{-\nu}$, within W , the expected number of points, $\mathbb{E}[W]$ is calculated below:

$$\mathbb{E}[W] = \int_W \lambda(x) dx = \int_0^R x^{-\nu} 2\pi x dx = \frac{2\pi}{\nu + 2} R^{\nu+2}$$

5.8.a

5.8.b

Figure 5.8: Comparison of the theoretical results of the first four moments per Theorem 5.4 and the corresponding Monte Carlo simulations under PPP type deployments: (a) Homogeneous PPP deployments. The parameters used: $R = 100, \alpha = 0.5, N = 10, \nu = \{0, 0.5\}$; (b) Non-homogeneous PPP deployments. The parameters used: $R = 100, \alpha = 0.5, \nu = \{0, 0.5\}$.

According to [128], pdf of x is shown as $g(x) = \frac{\lambda x}{\mathbb{E}[W]} = \frac{r^\nu(\nu+2)}{2\pi R^{\nu+2}}$. Given a small ε , the range of $g(x)$ is given by $\left[\frac{\nu+2}{2\pi R^2}, \frac{\varepsilon^\nu(\nu+2)}{2\pi R^{\nu+2}}\right]$. One way to generate the non-homogenous BPP is to use the accept and reject method. The procedure to generate NFBPP outlined below:

- (i) Generate N_s points within W , N_s should be large enough.
- (ii) Within these N_s , randomly selects N points with probability $\frac{\lambda(x)}{\varepsilon^c}$.

The value of ε will affect the approximation of theoretical moments. The smaller ε , the

better the approximation, however, the more difficult to generate the points inside the region. In the simulation we select $\varepsilon = 0.5$.

5.5.3 Comparison of Critical Region

When the sensors are located close to the center of the region, the empirical sample density and estimated density do not agree. This is because those points close to the center may violate the probability density because they will result in an inaccurate sample mean. Therefore, we remove a small hole around the center. We use r to represent the radius of the hole, which we call critical region. Figure 5.9.a shows the effect of removing those points for different critical region sizes. In particular, we vary $r \in \{0.1, 0.5, 1, 5\}$. We also compare the accuracy of series expansion methods with respect to different r . In fig. 5.9.b, we clearly show the effect that r has on the approximation for all three series expansion methods. In fig 5.9.b, we show the probability density function (PDF) for different values of r . From extensive numerical experiments, we have found that the choice $r = 5$ yields accurate approximations for a range of moments and will be used in all simulations.

5.5.4 Marginal Likelihood Estimation via Series Expansion and S-K Region

As discussed in Section 5.3, we use the Skew-Kurtosis curve to characterise regions where each of the series expansion would yield a positive PDF, given set of system parameters. It is therefore a useful tool to choose which series expansion to use. In this section, we present several examples to show how this will affect the approximation under each series expansion. The different sets of parameters are presented in Table 5.1.

5.5.4.1 Gram-Charlier series expansion

In Fig. 5.10 we plot the S-K curve and PDF Gram-Charlier series expansion estimation for three different sets of system parameters. The three sets of system parameters generate three different Skew-Kurtosis values, represented by the red dot in each of the figures. In the left figure, the S-K point falls at the region that satisfies the conditions in Lemma 5.3, while the other two do not satisfy the condition, which results in poor estimation of the PDF. This illustration shows the ability of the Gram-Charlier series expansion to accurately approximate the true distribution,

5.9.a

5.9.b

Figure 5.9: Effect of critical region on marginal likelihood estimation as a function of the critical region: (a) Four different sensor deployments with $r = \{0.1, 0.5, 1, 5\}$, $R = 100$, $\alpha = 2$, $\nu = 0$, $\sigma_w = 0$, $P_0 = 1$, $\theta = 2$, $\eta = 2$; (b) PDF estimation via Gram-Charlier, Gamma-Laguerre and Beta-Jacobi and Monte Carlo simulation.

depending on system parameters. It is clearly shown that for low SNR, the approximation is good, but for high SNR, the approximation has a negative PDF approximation.

5.5.4.2 Gamma-Laguerre series expansion

In Fig. 5.11 we plot the S-K curve and PDF Gram-Charlier series expansion estimation for three different sets of system parameters. In contrast with Gram-Charlier series expansion, it is clearly shown that for high SNR, the approximation is good. However, for low SNR, the

5.10.a

5.10.b

Figure 5.10: Gram-Charlier series expansion for three different system parameters presented in Table 5.1: (a) Skew-Kurtosis curves; (b) PDF estimation vs. Monte Carlo simulation.

Parameters	Gram Charlier			Gamma Laguerre			Beta Jacobi		
	Set 1	Set 2	Set 3	Set 1	Set 2	Set 3	Set 1	Set 2	Set 3
R	100	100	100	100	100	100	100	100	100
P_0	1	1	1	1	1	1	1	1	1
N	50	10	10	10	10	10	10	10	10
α	2	2	1.5	2.5	1.7	3	2.5	1.7	3
ν	0	0	0.6	0	0	0	0	0	0
σ_w	0	0	0.4	1	0	0.4	1	0	0.4

Table 5.1: Series Expansion Parameters

5.11.a

5.11.b

Figure 5.11: Gamma-Laguerre series expansion for three different system parameters in Table 5.1: (a) Skew-Kurtosis curves; (b) PDF estimation vs. Monte Carlo simulation.

approximation is not very accurate. This is because Gamma-Laguerre series expansion only has positive support.

5.5.4.3 Beta-Jacobi series expansion

In Fig. 5.12 we plot the S-K curve and PDF Beta-Jacobi series expansion estimation for three different sets of system parameters. For all three cases, the approximation is accurate. The only drawback for Beta-Jacobi estimation is the small fluctuations around the tail of the density.

Figure 5.12: Beta-Jacobi series expansion for three different system parameters in Table 5.1: (a) Skew-Kurtosis curves; (b) PDF estimation vs. Monte Carlo simulation.

5.5.5 Event Detection Performance Comparison

We now present the detection performance of the algorithms via Receiver Operating characteristics (ROC) for different series expansion methods. We select two representative scenarios of high and low SNR to show the direct effect of S-K curve has on the performance of the series expansions.

Fig. 5.13 presents the ROC curve for high SNR, $\sigma_w = 0.01$ and various values of the path-loss exponent $\alpha = \{2, 2.2, 2.4\}$. It can be seen from the ROC curves that Gram-Chalier expansion performs the worst while our Beta-Jacobi expansion performs the best in these three cases. The S-K curves under each case is also plotted. It can be seen from these figures that

the SK points are located outside the region for the Gram-Chalier expansion, while inside the region for Beta-Jacobi and Gamma-Laguerre expansions. The reason for the poor performance of Gram-Chalier is that the scenario is of high SNR values. When the additive noise variance is very small, it is difficult for Gram-Chalier to approximate the marginal likelihood under the alternative hypothesis since the support is quite narrow.

In Fig. 5.14, we present the ROC curve for low SNR, $\sigma_w = 0.4$ and various values of the path-loss exponent $\alpha = \{1.9, 2.1, 2.3\}$. In this case our Beta-Jacobi expansion outperforms the Gamma-Laguerre expansion and is comparable with the Gram-Chalier expansion. The system parameters are: $N = 20$, $\nu = 0.6$, $\sigma_w = 0.4$, $R = 100$. with α varying from 1.9 to 2.3. It is shown that Gamma-Laguerre performs the worst for these cases. This is because the support of the marginal density has negative values and Gamma-Laguerre can only have positive support. The same interpretation can also be found in the S-K curves.

5.6 Conclusions

We developed new event detection algorithms in Wireless Sensor Networks under two types of random spatial deployments. We formulated the problem as a binary hypothesis testing problem and designed optimal decision rule for it. The marginal densities were derived under both hypothesis. Low complexity and highly accurate series expansion methods were derived to approximate the marginal density under alternative hypothesis. The various series expansion methods were shown to be practical and suitable for different practical scenarios. We also show the event detection performance of Beta Jaocbi series expansion outperforms the other two methods for both high and low SNR cases. We also used extensive simulation results to generate the Receiver Operating Curves (ROC) under different sets of parameters. We verified the optimality of the series expansion methods and show the Skew-Kurtosis Curves for each method. Our schemes provided useful benchmark for other centralized and distributed scheme designs.

Figure 5.13: ROC curves for the three series expansions for different values of path-loss exponent $\alpha = \{2, 2.2, 2.4\}$.

Figure 5.14: ROC curves for the three series expansions for different values of path-loss exponent $\alpha = \{1.9, 2.1, 2.3\}$.

Chapter 6

Event Detection Algorithms with Incomplete Information from Sensors

This chapter presents event detection algorithms for WSNs, where sensors transmit incomplete information to GW. Section 6.1 presents event detection algorithm when information of sensors are transformed by Byzantine function. In Section 6.2, we consider event detection algorithms with nonlinear transmission function of each sensor. In section 6.3, extensive simulation results and discussions are presented for verifying the performance of the proposed algorithms. Section 6.4 concludes this chapter.

6.1 Event Detection Algorithms in WSNs with Byzantine Attacks

In this section, the event detection algorithm in WSNs with Byzantine attacks is presented. Specifically, in section 6.1.1, the sensor network model is proposed, followed by the optimal detection rule in section 6.1.2. Then the event detection algorithms for case I where CSI between source to sensors is unknown and from sensors to GW is perfect is presented in section 6.1.3. Then the event detection algorithm for case II where CSI between both source to sensors and sensors to GW are unknown is shown in section 6.1.4.

6.1.1 Sensor Network Model

We consider a wireless sensor network consisting of M sensors which observe an event represented by a single binary parameter $\theta \in \{0, 1\}$ over the wireless channel. GW knows the

value of θ as a priori. The system model is depicted in Fig. 6.1. Each sensor transmits its incoming signal to the gateway (GW). The incoming signal may be compromised by a malicious (Byzantine) attack, resulting in incomplete signal. The incompleteness of signal may be loss of information or falsification of signal which can cause the GW to make a false decision. The GW makes a binary centralized decision regarding the value of θ .

Figure 6.1: sensor network model with M sensors over wireless channels. Each sensor may be compromised by a malicious (Byzantine) attack $\mathcal{T}_B(R_m(l))$.

We now present the sensor network model:

- A1 The source is present $\theta=\theta$ ($\mathcal{H}_1^\theta$) or absent $\theta=0$ ($\mathcal{H}_0^\theta$) throughout a frame of L samples.
- A2 At each frame of L samples the observed signal at the m -th sensor ($m=1, \dots, M$) (denoted as $R_m(l)$) is given by:

$$\mathcal{H}_0^\theta : R_m(l) = V_m(l), l = 1, \dots, L$$

$$\mathcal{H}_1^\theta : R_m(l) = F_m(l)\theta + V_m(l), l = 1, \dots, L.$$

where $F_m(l)$ denotes the wireless channel between the source and the m -th sensor and $V_m(l) \sim CN(0, \sigma_V^2)$ is the i.i.d noise at the m -th sensor.

A3 Each of the M sensors may be either honest or Byzantine (malicious), where we assume an Independent Malicious Byzantine Attacks model [9]. In this case, each Byzantine sensor makes independent decisions based solely on its own observations. However, it is possible for a malicious user to launch other attacks, which we did not consider in the thesis. These can include the scenario where Byzantine sensors make dependent decisions not based solely on its own observation, but in collusion with other sensors based on their observations. We define $\Pr(\mathcal{H}_B^s)$ as the prior probability of a sensor to be Byzantine (malicious) given by:

$$\Pr(\mathcal{H}_B^s) = \frac{N}{M}, \quad (6.1)$$

where N is the average number of malicious sensors [114].

A4 **Sensor processing:** each sensor processes its observations before transmitting it to the GW as follows:

$$\begin{cases} \mathcal{H}_H^s : \mathcal{T}_H(R_m(l)) & (\text{Honest}), \\ \mathcal{H}_B^s : \mathcal{T}_B(R_m(l)) & (\text{Byzantine}), \end{cases}$$

where $\mathcal{T}_H : \mathbb{R} \rightarrow \mathbb{R}$ and $\mathcal{T}_B : \mathbb{R} \rightarrow \mathbb{R}$ are the relay functions of an honest sensor and a Byzantine sensor, respectively. A common example of $\mathcal{T}_H(R_m(l))$ is the Amplify-and-Forward function [58]. The function of the malicious sensor $\mathcal{T}_B(R_m(l))$ will be detailed in Section 6.1.2.1.

A5 The received signal at the GW from m -th sensor over wireless channels at epoch l is given by

$$Y_m(l) = G_m(l)\mathcal{T}_i(R_m(l)) + W(l), \quad i \in \{H, B\}. \quad (6.2)$$

$y_m(l)$ is obtained in the following model:

$$y_m(l) = \begin{cases} G_m(l)\mathcal{T}_H(V_m(l)) + W(l), & \mathcal{H}_H^s, \mathcal{H}_0^\theta \\ G_m(l)\mathcal{T}_H(F_m(l)\theta + V_m(l)) + W(l), & \mathcal{H}_H^s, \mathcal{H}_1^\theta \\ G_m(l)\mathcal{T}_B(V_m(l)) + W(l), & \mathcal{H}_B^s, \mathcal{H}_0^\theta \\ G_m(l)\mathcal{T}_B(F_m(l)\theta + V_m(l)) + W(l), & \mathcal{H}_B^s, \mathcal{H}_1^\theta \end{cases} \quad (6.3)$$

where $y_m(l)$ is the received signal at the l -th sample for m -th sensor, $G_m(l)$ is the wireless channel between m -th sensor and the GW. $F_m(l)$ is the wireless channel between the source and m -th sensor. The additive noise at the GW is $W(l) \sim CN(0, \sigma_W^2)$, and $V_m(l)$ is the random additive noise at the m -th sensor.

A6 All wireless channels are assumed to be independent and follow a Rayleigh distribution (denoted as CN, i.e., complex normal distribution), as follows:

$$\begin{aligned}\mathbf{F}(l) &\sim CN(\overline{\mathbf{F}}(l), \Sigma_{\mathbf{F}}), & (\text{Source} \rightarrow \text{Sensor links}) \\ \mathbf{G}(l) &\sim CN(\overline{\mathbf{G}}(l), \Sigma_{\mathbf{G}}), & (\text{Sensors} \rightarrow \text{FC links})\end{aligned}$$

where $\mathbf{F}(l) \in C^{M \times 1}$ is the wireless channel between the source and sensors and $\mathbf{G}(l) \in C^{M \times 1}$ is the wireless channel matrix between the sensors and the GW. $\overline{\mathbf{G}}(l)$ and $\overline{\mathbf{F}}(l)$ are the channels mean values, and $\Sigma_{\mathbf{F}} = \sigma_{\mathbf{F}}^2 \mathbf{I}$ and $\Sigma_{\mathbf{G}} = \sigma_{\mathbf{G}}^2 \mathbf{I}$ are the covariance matrix.

6.1.2 Optimal Detection Decision Rule

In this section we derive the optimal detection decision rule. The objective of the decision rule is to identify which hypothesis an observation belongs to, given a set of observations. We develop an algorithm to perform the optimal decision rule.

The decision rule is a threshold test based on the likelihood ratio [116], given by:

$$\Lambda(\mathbf{Y}(1:L)) \triangleq \frac{p(\mathbf{y}_{1:M}(1:L)|\mathcal{H}_1^\theta)}{p(\mathbf{y}_{1:M}(1:L)|\mathcal{H}_0^\theta)} \underset{\mathcal{H}_0^\theta}{\overset{\mathcal{H}_1^\theta}{\geq}} \gamma, \quad (6.4)$$

where the threshold γ can be set to assure a fixed system false-alarm rate under the Neyman-Pearson approach or can be chosen to minimize the overall probability of error under the Bayesian approach [124].

We can further decompose the full marginals under each hypothesis, $p(\mathbf{y}_{1:M}(1:L)|\mathcal{H}_k^\theta)$, $k = 0, 1$, and marginalise over the sensor's state (Honest, Byzantine) as follows:

$$\begin{aligned}p(\mathbf{y}_{1:M}(1:L)|\mathcal{H}_k^\theta) &= \prod_{l=1}^L p(\mathbf{y}_{1:M}(l)|\mathcal{H}_k^\theta) \\ &= \prod_{l=1}^L \prod_{m=1}^M p(y_m(l)|\mathcal{H}_k^\theta) \\ &= \prod_{l=1}^L \prod_{m=1}^M \sum_{j \in \{H, B\}} p(y_m(l)|\mathcal{H}_j^s, \mathcal{H}_k^\theta) \Pr(\mathcal{H}_j^s).\end{aligned} \quad (6.5)$$

Before we present our algorithms for calculating the optimal decision rule in (6.5) we need to define the Byzantine attack function $\mathcal{T}_B(\cdot)$

6.1.2.1 Byzantine attack function optimization

In this section we derive the optimal Byzantine attack function to increase the false alarm rate from the attacker's point of view. A compromised (malicious) sensor will transform the observation in such a way that it would seem to have been generated from the opposite hypothesis, therefore fooling the GW. This means that the Byzantine function should attempt to satisfy the following:

$$\begin{aligned}\mathcal{T}_{B,m}(R_m(l)|\mathcal{H}_0^\theta) &\stackrel{d}{=} CN(\overline{F}_m(l)\theta, \theta^2\sigma_F^2 + \sigma_V^2) \\ \mathcal{T}_{B,m}(R_m(l)|\mathcal{H}_1^\theta) &\stackrel{d}{=} CN(0, \sigma_V^2),\end{aligned}$$

where $\stackrel{d}{=}$ denotes equivalence between the distributions. To execute this strategy perfectly, the attacker would need to estimate the value of θ . However, the attacker may choose to implement a linear attack function $\mathcal{T}_{B,m} = a_mx + b_m$ for the sake of simplicity. We say that the function is optimal if it minimizes the sum of the Fréchet distances [129, 130] between $R_m(l)|\mathcal{H}_i^\theta$ and $\mathcal{T}_{B,m}(R_m(l)|\mathcal{H}_i^\theta)$ for each $i \in \{0, 1\}$. In Lemma 6.1, we present the Byzantine attack function.

Lemma 6.1. *The linear Byzantine attack function is given by $\mathcal{T}_{B,m}(x) = a_mx + b_m$, where*

$$\begin{bmatrix} a_m \\ b_m \end{bmatrix} = (P^T P)^{-1} P^T Q,$$

where

$$\begin{aligned}P &= \begin{pmatrix} \sigma_V & 0 \\ \sqrt{\theta^2\sigma_F^2 + \sigma_V^2} & 0 \\ 0 & 1 \\ \overline{F}_m(l)\theta & 1 \end{pmatrix}, \\ Q &= \begin{pmatrix} \sqrt{\theta^2\sigma_F^2 + \sigma_V^2} \\ \sigma_V \\ \overline{F}_m(l)\theta \\ 0 \end{pmatrix}.\end{aligned}$$

Proof. Given two normal variables, $R_m(l)|\mathcal{H}_0^\theta \sim \mathcal{N}(0, \sigma_V^2)$ and $R_m(l)|\mathcal{H}_1^\theta \sim \mathcal{N}(\overline{F}_m(l)\theta, \theta^2\sigma_F^2 + \sigma_V^2)$, the optimality condition on $\mathcal{T}_{B,m}$ is equivalent to minimizing $[(a \cdot 0 + b) - \overline{F}_m(l)\theta]^2 + [(a \cdot \overline{F}_m(l) + b) - 0]^2 + [(a\sigma_V) - \sqrt{\theta^2\sigma_F^2 + \sigma_V^2}]^2 + [a\sqrt{\theta^2\sigma_F^2 + \sigma_V^2} - \sigma_V]^2$. We solve this optimization problem via least squares by minimizing $|Q - P\tilde{z}|^2$, $\tilde{z} = [a, b]^T$, which results in the solution $\hat{\tilde{z}} = (P^T P)^{-1} P^T Q$. $\square$

6.1.3 Unknown CSI Between Source to Sensors and Perfect CSI Between Sensors to GW

We derive the optimal decision rule in (6.4) for the case in which the GW has perfect knowledge of $\mathbf{G}(l)$ and no knowledge of $\mathbf{F}(l)$. This scenario is practical in cases where training phase for channel estimation between the sensors and the GW is available.

The marginal likelihood of the system model (6.3), when $\mathbf{G}(l)$ is known and $\mathbf{F}(l)$ is random unknown, is given by:

$$y_m(l)|\mathbf{g}(l) \sim \begin{cases} CN(\underbrace{0}_{\mu_{H0,m}(l)}, \underbrace{\sigma_V^2 \alpha^2 g_m(l) g_m(l)^H + \sigma_W^2}_{\Sigma_{H0,m}(l)}), & \mathcal{H}_H^s, \mathcal{H}_0^\theta \\ CN(\underbrace{\alpha g_m(l) \bar{F}_m(l) \theta}_{\mu_{H1,m}(l)}, \underbrace{\alpha^2 \beta(l) + \sigma_W^2}_{\Sigma_{H1,m}(l)}), & \mathcal{H}_H^s, \mathcal{H}_1^\theta \\ CN(\underbrace{b g_m(l)}_{\mu_{B0,m}(l)}, \underbrace{a^2 \sigma_V^2 g_m(l) g_m(l)^H + \sigma_W^2}_{\Sigma_{B0,m}(l)}), & \mathcal{H}_B^s, \mathcal{H}_0^\theta \\ CN(\underbrace{a g_m(l) \bar{F}_m(l) \theta + b g_m(l)}_{\mu_{B1,m}(l)}, \underbrace{a^2 \beta(l) + \sigma_W^2}_{\Sigma_{B1,m}(l)}), & \mathcal{H}_B^s, \mathcal{H}_1^\theta \end{cases} \quad (6.6)$$

where $\beta(l) \triangleq (\sigma_V^2 + \theta^2 \sigma_F^2) g_m(l) g_m(l)^H$ and α is the amplify-and-forward coefficient. (6.6) shows the Rayleigh distribution of system model (6.3), where $\mu_{Hi,m}(l), \Sigma_{Hi,m}(l), i = 0, 1$ refer to the mean and deviation for $\mathcal{H}_H^s, \mathcal{H}_i^\theta$ and $\mu_{Bi,m}(l), \Sigma_{Bi,m}(l), i = 0, 1$ for $\mathcal{H}_B^s, \mathcal{H}_i^\theta$ respectively.

By using the decomposition (6.5), the test statistic in (6.4) is given by:

$$\begin{aligned} \Lambda(\mathbf{Y}(1:L)) &\triangleq \prod_{l=1}^L \prod_{m=1}^M \frac{p(y_m(l)|\mathcal{H}_1^\theta)}{p(y_m(l)|\mathcal{H}_0^\theta)} \\ &= \prod_{l=1}^L \prod_{m=1}^M \frac{p(y_m(l)|\mathcal{H}_H^s, \mathcal{H}_1^\theta) \Pr(\mathcal{H}_H^s) + p(y_m(l)|\mathcal{H}_B^s, \mathcal{H}_1^\theta) \Pr(\mathcal{H}_B^s)}{p(y_m(l)|\mathcal{H}_H^s, \mathcal{H}_0^\theta) \Pr(\mathcal{H}_H^s) + p(y_m(l)|\mathcal{H}_B^s, \mathcal{H}_0^\theta) \Pr(\mathcal{H}_B^s)} \end{aligned} \quad (6.7)$$

where $\Pr(\mathcal{H}_B^s)$ is defined in A3, $\Pr(\mathcal{H}_H^s) = 1 - \Pr(\mathcal{H}_B^s)$ and $p(y_m(l)|\mathcal{H}_j^s, \mathcal{H}_i^\theta)$ is given by:

$$p(y_m(l)|\mathcal{H}_j^s, \mathcal{H}_i^\theta) = \frac{\exp^{-\frac{1}{2}(y_m(l) - \mu_{ji}(l))^H \Sigma_{ji}^{-1}(l) (y_m(l) - \mu_{ji}(l))}}{\sqrt{2\pi \Sigma_{ji}(l)}} \quad (6.8)$$

Algorithm 6.1 Event Detection Algorithm for Case I

Input: $\mathbf{Y}_{1:M}(1 : L)$, γ , $\bar{\mathbf{F}}$, $\mathbf{G}$, $\Sigma_{\mathbf{F}}$, $\Sigma_{\mathbf{V}}$, $\Sigma_{\mathbf{W}}$
Output: Binary decision $(\mathcal{H}_0^\theta, \mathcal{H}_1^\theta)$

- (i) Calculate $\Pr(\mathcal{H}_j^s)$ according to (6.1).
 - (ii) Calculate $p(y_m(l)|\mathcal{H}_j^s, \mathcal{H}_i^\theta)$ according to (6.8).
 - (iii) Calculate $\Lambda(\mathbf{Y}_{1:M}(1 : L))$ via (6.7) and compare with the threshold γ .
-

where $\mu_{ji}(l)$ and $\Sigma_{ji}(l)$ are given in (6.6). We take the logarithm of (6.7) and get:

$$\begin{aligned} \log\{\Lambda(\mathbf{Y}(1 : L))\} \triangleq & \sum_{l=1}^L \sum_{m=1}^M \{ \log(p(y_m(l)|\mathcal{H}_H^s, \mathcal{H}_1^\theta) \Pr(\mathcal{H}_H^s) \\ & + p(y_m(l)|\mathcal{H}_B^s, \mathcal{H}_1^\theta) \Pr(\mathcal{H}_B^s)) \} - \log\{p(y_m(l)|\mathcal{H}_H^s, \mathcal{H}_0^\theta) \Pr(\mathcal{H}_H^s) \\ & + p(y_m(l)|\mathcal{H}_B^s, \mathcal{H}_0^\theta) \Pr(\mathcal{H}_B^s)) \}. \end{aligned} \quad (6.9)$$

The false alarm probability and positive detection probability are given by $p(\Lambda(\mathbf{Y}(1 : L)) > \gamma|\mathcal{H}_0^\theta)$ and $p(\Lambda(\mathbf{Y}(1 : L)) > \gamma|\mathcal{H}_1^\theta)$. Deriving these probabilities involves intractable integrals which cannot be expressed in closed form. We therefore perform Monte Carlo simulations to show the effect of parameters on these probabilities. The event detection algorithm is shown in Algorithm 6.1.

6.1.4 Unknown CSI Between Source to Sensors and Unknown CSI Between Sensors to GW

We generalize the case in Section 6.1.3 by considering the scenario where there is no knowledge of both $\mathbf{G}(l)$ and $\mathbf{F}(l)$. This case is applicable when there is no training phase for estimating the channels between the sensors and the GW. Consequently, the distribution of the marginal likelihood $p(y_m(l)|\mathcal{H}_k^\theta)$, $k = 0, 1$ in (6.2) is intractable.

To overcome this problem, we develop a novel low-complexity detection algorithm based on moment matching. This algorithm approximates one distribution with another distribution, by matching their moments. A popular choice is to match the distribution with a normal distribution, due to its simplicity. Our approximation is based on Lemma 6.2.

Lemma 6.2. *The first two moments of $Z=XY$, where $X \sim N(\overline{G_m}(l), \sigma_{G_m(l)}^2)$ and $Y \sim N(\mathcal{T}_i(\overline{R_m}(l)), \sigma_{\mathcal{T}_i(R_m(l))}^2)$ is given by:*

$$\mathbb{E}[Z] = \mathbb{E}[XY] = \overline{G_m}(l) \cdot \mathcal{T}_i(\overline{R_m}(l))$$

$$\begin{aligned} \text{Var}(Z) &= \mathbb{E}[X^2]\mathbb{E}[Y^2] - (\mathbb{E}[XY])^2 \\ &= (\sigma_{G_m(l)}^2 + \overline{G_m}(l)^2)(\sigma_{\mathcal{T}_i(R_m(l))}^2 + \mathcal{T}_i(\overline{R_m}(l))^2) \\ &\quad - \overline{G_m}(l)^2 \mathcal{T}_i(\overline{R_m}(l))^2 \end{aligned}$$

To apply Lemma 6.2, we first observe that the distribution of $\mathcal{T}_i(R_m(l))$ is given by:

$$\left\{ \begin{array}{ll} \underbrace{CN(0, \alpha^2 \sigma_V^2)}_{\Sigma_0}, & \mathcal{H}_H^s, \mathcal{H}_0^\theta \\ \underbrace{CN(\underbrace{\alpha \theta \overline{F}_m(l)}_{\Sigma_0}, \underbrace{\alpha^2 \theta^2 \sigma_F^2 + \sigma_V^2}_{\Sigma_1})}_{\Sigma_1}, & \mathcal{H}_H^s, \mathcal{H}_1^\theta \\ \underbrace{CN(b, a^2 \sigma_V^2)}_{\Sigma_2}, & \mathcal{H}_B^s, \mathcal{H}_0^\theta \\ \underbrace{CN(\underbrace{a \theta \overline{F}_m(l)}_{\mu_3} + b, \underbrace{a^2 \theta^2 \sigma_F^2 + a^2 \sigma_V^2}_{\Sigma_3})}_{\Sigma_3}, & \mathcal{H}_B^s, \mathcal{H}_1^\theta \end{array} \right.$$

Consequently, the distribution of the received signal in (6.10) will be approximated by normal distributions:

$$y_m(l) \stackrel{d}{\sim} \left\{ \begin{array}{l} \underbrace{CN(0, (\sigma_G^2 + \overline{G_m}(l)^2)\Sigma_0 + \sigma_W^2)}_{\mathcal{H}_H^s, \mathcal{H}_0^\theta} \\ \underbrace{CN(\overline{G_m}(l)\mu_1, (\sigma_G^2 + \overline{G_m}(l)^2)(\Sigma_1 + \mu_1^2) - \overline{G_m}(l)^2\mu_1^2 + \sigma_W^2)}_{\mathcal{H}_H^s, \mathcal{H}_1^\theta} \\ \underbrace{CN(b\overline{G_m}(l), (\sigma_G^2 + \overline{G_m}(l)^2)(\Sigma_2 + b^2) - \overline{G_m}(l)^2b^2 + \sigma_W^2)}_{\mathcal{H}_B^s, \mathcal{H}_0^\theta} \\ \underbrace{CN(\overline{G_m}(l)\mu_3, (\sigma_G^2 + \overline{G_m}(l)^2)(\Sigma_3 + \mu_3^2) - \overline{G_m}(l)^2\mu_3^2 + \sigma_W^2)}_{\mathcal{H}_B^s, \mathcal{H}_1^\theta} \end{array} \right.$$

where $\stackrel{d}{\sim}$ denotes approximation in distribution. The distribution of $y_m(l)$ has the same structure as (6.6). Therefore the detection algorithm under Gaussian approximation is implemented similarly to Algorithm 6.1.

6.2 Event Detection Algorithms in WSNs with Nonlinear Transmission

In this section, the event detection algorithms in WSNs with nonlinear transmission is shown. Nonlinear transmission happens when there is nonlinear amplifying in sensors. The system model and definitions are presented in section 6.2.1. The series expansion methods using Beta-Jacobi are presented in section 6.2.2. Then we calculate the moments in section 6.2.3.

6.2.1 Wireless Sensor Network System Model and Definitions

We consider a wireless sensor network consisting of M sensors which observe an event represented by a single binary parameter $\theta \in \{0, \theta\}$ over the wireless channel. GW knows the value of θ as a-priori. The system model is depicted in Fig. 6.2. Each sensor transmits its incoming signal to the gateway (GW). The GW makes a binary centralized decision regarding the value of θ .

We now present the sensor network model:

Figure 6.2: Wireless sensor network with M sensors over wireless channels.

- A1 The source is present $\theta=\theta$ ($\mathcal{H}_1^\theta$) or absent $\theta=0$ ($\mathcal{H}_0^\theta$) throughout a frame of L samples.
- A2 At each frame of L samples the observed signal at the m -th sensor ($m=1, \dots, M$) (denoted as $R_m(l)$) is given by:

$$\begin{cases} R_m(l) = V_m(l), & \mathcal{H}_0 \\ R_m(l) = H_m(l)\theta + V_m(l), & \mathcal{H}_1. \end{cases}$$

where $H_m(l)$ denotes the wireless channel between the source and the m -th sensor and $V_m(l) \sim N(0, \sigma_V^2)$ is the i.i.d noise at the m -th sensor.

- A3 **Sensor processing:** each sensor uses the amplify-and-forward (AF) strategy to transmit its observations [58]. Due to power constraints and non-linearity of the amplifier, the sensor's amplifier may be saturated, leading to the following mixture model

$$\mathcal{F}_R(\alpha) = \begin{cases} \mathcal{F}_L(\alpha), & \text{w.p } p_l \\ \mathcal{F}_{NL}(\alpha), & \text{w.p } (1 - p_l) \end{cases}$$

where $\mathcal{F}_L(\alpha)$ and $\mathcal{F}_{NL}(\alpha)$ are the linear and non-linear relay amplification functions, respectively, and p_l is the a-priori probability of using the linear function.

The linear AF function is given by

$$\mathcal{F}_L(\alpha) = g\alpha,$$

where g is a known amplification factor. The non-linear amplification function

$$\mathcal{F}_{NL}(\alpha) = \begin{cases} KP_{\max}, & \alpha \geq P_{\max}, \\ K\alpha, & P_{\min} \leq \alpha \leq P_{\max}, \\ KP_{\min}, & \alpha \leq P_{\min}. \end{cases}$$

- A4 The received signal at the GW from m -th sensor over wireless channels at epoch l is given by the following model:

$$y_m(l) = \begin{cases} G_m(l) \mathcal{F}_R(V_m(l)) + W(l), & \mathcal{H}_0 \\ G_m(l) \mathcal{F}_R(H_m(l)\theta + V_m(l)) + W(l), & \mathcal{H}_1 \end{cases} \quad (6.10)$$

where $y_m(l)$ is the received signal at the l -th sample for m -th sensor, $G_m(l)$ is the wireless channel between m -th sensor and the GW. $H_m(l)$ is the wireless channel between the source and m -th sensor. The additive noise at the GW is $W(l) \sim N(0, \sigma_W^2)$, and $V_m(l)$ is the random additive noise at the m -th sensor.

A5 All wireless channels are assumed to be independent and follow a Rayleigh distribution (denoted as N , i.e., normal distribution), as follows:

$$\begin{aligned}\mathbf{H}(l) &\sim N(\overline{\mathbf{H}}(l), \Sigma_{\mathbf{H}}), \quad (\text{Source} \rightarrow \text{Sensor links}) \\ \mathbf{G}(l) &\sim N(\overline{\mathbf{G}}(l), \Sigma_{\mathbf{G}}), \quad (\text{Sensors} \rightarrow \text{FC links})\end{aligned}$$

where $\mathbf{H}(l)$ is the wireless channel between the source and sensors and $\mathbf{G}(l)$ is the wireless channel matrix between the sensors and the GW. $\overline{\mathbf{G}}(l)$ and $\overline{\mathbf{H}}(l)$ are the channel mean values, and $\Sigma_{\mathbf{F}} = \sigma_{\mathbf{F}}^2 \mathbf{I}$ and $\Sigma_{\mathbf{G}} = \sigma_{\mathbf{G}}^2 \mathbf{I}$ are the covariance matrix.

6.2.1.1 Problem formulation: optimal detection decision rule

The decision rule is a threshold test based on the likelihood ratio [116], given by:

$$\Lambda(\mathbf{Y}(1:L)) \triangleq \frac{p(\mathbf{y}_{1:M}(1:L) | \mathcal{H}_1)}{p(\mathbf{y}_{1:M}(1:L) | \mathcal{H}_0)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \gamma, \quad (6.11)$$

We can decompose the marginal likelihood under each model as follows:

$$\begin{aligned}p(\mathbf{y}_{1:M}(1:L) | \mathcal{H}_j) &= \prod_{l=1}^L p(\mathbf{y}_{1:M}(l) | \mathcal{H}_j) \\ &= \prod_{l=1}^L \prod_{m=1}^M p(y_m(l) | \mathcal{H}_j).\end{aligned} \quad (6.12)$$

We now express $p(y_m(l))$ under each hypothesis:

$$\begin{aligned}p(y_m(l) | \mathcal{H}_0) &= \int p(y_m(l) | \mathcal{F}_{\mathbf{R}}(R_m(l))) p(\mathcal{F}_{\mathbf{R}}(R_m(l))) d\mathcal{F}_{\mathbf{R}}(R_m(l)) \\ &= \underbrace{p_l \int p(y_m(l) | \mathcal{F}_{\mathbf{R}}(R_m(l)) = \mathcal{F}_{\mathbf{L}}(V_m(l))) d(\mathcal{F}_{\mathbf{L}}(V_m(l)))}_{\text{Linear relay}} \\ &\quad + \underbrace{(1 - p_l) \int p(y_m(l) | \mathcal{F}_{\mathbf{R}}(R_m(l)) = \mathcal{F}_{\mathbf{NL}}(V_m(l))) d(\mathcal{F}_{\mathbf{NL}}(V_m(l)))}_{\text{Non-linear relay}}. \\ p(y_m(l) | \mathcal{H}_1) &= \int p(y_m(l) | \mathcal{F}_{\mathbf{R}}(R_m(l))) p(\mathcal{F}_{\mathbf{R}}(R_m(l))) d\mathcal{F}_{\mathbf{R}}(R_m(l)) \\ &= \underbrace{p_l \int p(y_m(l) | \mathcal{F}_{\mathbf{R}}(R_m(l)) = \mathcal{F}_{\mathbf{L}}(H_m(l)\theta + V_m(l))) d(\mathcal{F}_{\mathbf{L}}(H_m(l)\theta + V_m(l)))}_{\text{Linear relay}} \\ &\quad + \underbrace{(1 - p_l) \int p(y_m(l) | \mathcal{F}_{\mathbf{R}}(R_m(l)) = \mathcal{F}_{\mathbf{NL}}(H_m(l)\theta + V_m(l))) d(\mathcal{F}_{\mathbf{NL}}(H_m(l)\theta + V_m(l)))}_{\text{Non-linear relay}}.\end{aligned} \quad (6.13)$$

Unfortunately, the marginal distribution under both models cannot be calculated analytically, prohibiting direct implementation of the detection rule in (6.11). To approximate the marginal likelihood Y_l we will utilise a series expansion approach presented in the next section.

6.2.2 Event Detection via Beta-Jacobi Density Series Expansion

In order to evaluate the marginal likelihood in (6.13), we adopt the novel approximation which was proposed in section 5.3.3. The calculation of the series expansion requires deriving the moment of Y under both hypothesis, presented in the next Section.

6.2.3 Calculation of The Moments

One approach to calculating the moments is by obtaining the Moment Generating Function (MGF) of $Y_m(l)$, given by:

$$M_{Y_m}(t) = \mathbb{E}_{f_{Y_m}} [\exp(ty)] = \int p(y_m(l)) \exp(ty) dy,$$

Since direct calculation of this integral is difficult, we derive instead the MGF by calculating the p -th moment for $y_m(l)$ and then deriving the MGF based on the moments. Calculating the p -th moment involves the following steps:

- (i) Calculate the k -th moment of $\mathcal{F}_R(R_m(l))$.
- (ii) Calculate the k -th moment of $G_m(l)\mathcal{F}_R(R_m(l))$.
- (iii) Calculate MGF of $Z_m(l) = G_m(l)\mathcal{F}_R(R_m(l))$.
- (iv) Calculate MGF of $Y_m(l)$.
- (v) Calculate the moments of $Y_m(l)$.
- (vi) Use beta Jacobi series expansions to find $p(Y_m(l))$.
- (vii) Calculate $\Lambda(Y_m(l))$ and compare with threshold γ .

Theorem 6.1. *The k^{th} moment for $\mathcal{F}_R(R_m(l))$ is given by:*

$$\mathbb{E} \left[\mathcal{F}_R(R_m(l))^k \right] = \begin{cases} p_l \mathbb{E} \left[\mathcal{F}_L(V_m(l))^k \right] + (1 - p_l) \mathbb{E} \left[\mathcal{F}_{NL}(V_m(l))^k \right], & \mathcal{H}_0 \\ p_l \mathbb{E} \left[\mathcal{F}_L(H_m(l)\theta + V_m(l))^k \right] + (1 - p_l) \mathbb{E} \left[\mathcal{F}_{NL}(H_m(l)\theta + V_m(l))^k \right], & \mathcal{H}_1 \end{cases} \quad (6.14)$$

The moments under linear relay function are given by:

$$\mathbb{E} \left[\mathcal{F}_L(\cdot)^k | \mathcal{H}_j \right] = \sigma_j^k \left(-i\sqrt{2} \text{sgn}(\mu_j) \right)^k U \left(-\frac{1}{2}k, \frac{1}{2}, -\frac{1}{2}(\mu_j/\sigma_j)^2 \right) \quad j \in \{0, 1\},$$

where $\mu_0 = 0$, $\mu_1 = g\bar{F}\theta$, $\sigma_0^2 = g^2\sigma_V^2$ and $\sigma_1^2 = g^2\theta^2\sigma_V^2 + g^2\sigma_H^2$.

The moments under non-linear relay function are given by:

$$\mathbb{E} [\mathcal{F}_{NL}(\cdot) | \mathcal{H}_j] = K^k P_{\min}^k \Phi \left(\frac{P_{\min} - \mu_j}{\sigma_j} \right) + (I_{P_{\max}} - I_{P_{\min}}) + K^k P_{\max}^k \left(1 - \Phi \left(\frac{P_{\max} - \mu_j}{\sigma_j} \right) \right),$$

where $\mu_0 = 0$, $\mu_1 = \bar{F}\theta$, $\sigma_0^2 = \sigma_V^2$ and $\sigma_1^2 = \theta^2\sigma_V^2 + \sigma_H^2$.

In addition, we denote by $\phi(\cdot)$ and $\Phi(\cdot)$ the standard Normal probability density function and cumulative density function, $I_{P_{\max}}$ and $I_{P_{\min}}$ are defined in Appendix A.7, $U(a, b, z)$ is the Tricomi's (confluent hypergeometric) function and $\text{sgn}(x)$ is the signum function.

Proof. See Appendix A.7. □

Next, the k^{th} moment of $Z_m(l) := G_m(l) \mathcal{F}_R(R_m(l))$ can be easily calculated as a product due to their independence:

$$\mathbb{E} \left[Z_m(l)^k \right] = \mathbb{E} \left[G_m(l)^k \right] \mathbb{E} \left[\mathcal{F}_R(R_m(l))^k \right].$$

Next, the MGF of $Z_m(l)$ is calculated

$$M_Z(t) = 1 + \sum_{k=1}^{\infty} \frac{t^k \mathbb{E} [Z_m(l)^k]}{k!}. \quad (6.15)$$

Due to the independence of $Z_m(l)$ and $w(l)$, the moment generating function for $y_m(l)$ is given by:

$$M_Y(t) = M_Z(t) M_W(t) = \left(1 + \sum_{k=1}^{\infty} \frac{t^k}{k!} \mathbb{E} [Z_m(l)^k] \right) e^{\frac{1}{2}\sigma_W^2 t^2}$$

Algorithm 6.2 Event Detection Algorithm

Input: $Y_l, \gamma, \bar{F}, \sigma_F, \bar{G}, \sigma_G, \sigma_v, \sigma_w$
Output: Binary decision ($\mathcal{H}_1, \mathcal{H}_0$)

- (i) Calculate κ_i and μ_i .
 - (ii) Evaluate the series expansion chosen in Section 6.2.2 to find $p(\mathbf{y}_{1:M}(1:L)|\mathcal{H}_k^\theta)$.
 - (iii) Calculate $\Lambda(Y_l)$ via (6.11) and compare to the threshold γ .
-

In order to derive cumulants and moments, first we find the cumulant function:

$$g(t) = \log M_Y(t) = \log \left(1 + \sum_{k=1}^{\infty} \frac{t^k}{k!} \mathbb{E}[Z_m(l)^k] \right) + \frac{1}{2} \sigma_w^2 t^2,$$

Next the first cumulant κ_1 is given by:

$$\begin{aligned} \kappa_1 &= \left. \frac{dg(t)}{dt} \right|_{t=0} \\ &= \left. \frac{\frac{t^0}{1} \mathbb{E}[Z_m(l)] + \sum_{k=2}^{\infty} k \frac{t^{k-1}}{k!} \mathbb{E}[Z_m(l)^k]}{1 + \sum_{k=1}^{\infty} \frac{t^k}{k!} \mathbb{E}[Z_m(l)^k]} \right|_{t=0} + \sigma_w^2 t|_{t=0} \\ &= \mathbb{E}[Z_m(l)] \end{aligned}$$

Similarly, by taking the second, third and fourth derivative of $g(t)$, we calculate κ_2, κ_3 and κ_4 :

$$\kappa_1 = \mathbb{E}[Z_m(l)],$$

$$\kappa_2 = \mathbb{E}[Z_m(l)^2] - (\mathbb{E}[Z_m(l)])^2 + \sigma_w^2,$$

$$\kappa_3 = \mathbb{E}[Z_m(l)^3] - 3\mathbb{E}[Z_m(l)^2] \mathbb{E}[Z_m(l)] + 2(\mathbb{E}[Z_m(l)])^3,$$

$$\kappa_4 = \mathbb{E}[Z_m(l)^4] - 4\mathbb{E}[Z_m(l)^3] \mathbb{E}[Z_m(l)] - 3(\mathbb{E}[Z_m(l)^2])^2 + 12\mathbb{E}[Z_m(l)^2] (\mathbb{E}[Z_m(l)])^2 - 6(\mathbb{E}[Z_m(l)])^4$$

The Moments can be expressed as polynomials of cumulants:

$$\mu_1 = \kappa_1,$$

$$\mu_2 = \kappa_2 + \kappa_1^2,$$

$$\mu_3 = \kappa_3 + 3\kappa_2\kappa_1 + \kappa_1^3,$$

$$\mu_4 = \kappa_4 + 4\kappa_3\kappa_1 + 3\kappa_2^2 + 6\kappa_2\kappa_1^2 + \kappa_1^4.$$

6.3 Simulation Results

In this section, we present the simulation results for both algorithms derived in section 6.1 and section 6.2.

6.3.1 Event Detection Algorithm with Byzantine Attack

In this section, we present Monte Carlo simulation results for the proposed algorithms in the two cases of CSI of Sections 6.1.3 and 6.1.4. The setting for all the simulations are as follows: the prior distribution for all the channels is Rayleigh fading and the channels are assumed to be both spatially and temporally independent. We set the observed binary parameter, θ to 1 and the results are obtained from simulations over 50,000 realizations of channel and noise for a given set of N, M, L, σ_V^2 and σ_W^2 . In the following sections we present the detection performance of the algorithms via Receiver Operating Characteristics (ROC) for the different configurations: number of sensors, $M = 100$. Ratio of malicious users, $\Pr(\mathcal{H}_B^s) = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.55, 0.6\}$, number of samples, $L = 1$ and noise variance, $\sigma_V^2 = \sigma_W^2 = 0.25, \alpha = 1$.

6.3.1.1 Case I-Unknown CSI between source and sensors and Perfect CSI between Sensors and GW

Fig. 6.3 presents ROC results for the case of unknown CSI between the source and sensors and perfect CSI between sensors and the GW (Section 6.1.3) for the optimal detection algorithm presented in Algorithm 6.1. The results show that with the increase in number of malicious users, the performance of the algorithm decreases. This implies the optimization of malicious function in Section 6.1.2.1 affects the performance of algorithm, as intended. When the number of malicious users is 0, the algorithm has the highest positive detection probabilities. As the ratio of malicious users grows, it becomes increasingly difficult for the GW to determine whether the data sent from sensor is malicious. Interestingly, for false detection rates of less than 20 %, the system can tolerate up to 20% malicious users while maintaining a positive detection probability higher than 80%.

Fig. 6.4 presents ROC results for case I as the mean $\bar{F}$ of the unknown channel between the source and the sensors varies. A larger mean could mean a stronger line of sight between

Figure 6.3: Case I - ROC performance given unknown CSI between the source and sensors and perfect CSI between sensors and the GW under different configurations of the number of malicious sensors $\Pr(\mathcal{H}_B^s) \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.55, 0.6\}$, ($M = 100, L = 1$)

Figure 6.4: Case I - ROC performance given unknown CSI between the source and sensors and perfect CSI between sensors and the GW under different configurations of $\bar{\mathbf{F}} \in \{0.1 + 0.1i, \dots, 0.5 + 0.5i\}$, ($\Pr(\mathcal{H}_B^s) = 0.3, M = 100, L = 1$)

sources and sensors, and stronger line of sight leads to a better performance for the algorithm. The reason is that when $\bar{\mathbf{F}}$ increases, the difference between the $\mathcal{H}_H^s(0)$ and $\mathcal{H}_B^s(\theta)$ also increases, making it easier for the GW to find out which hypothesis the observation belongs

Figure 6.5: Case II - ROC performance under unknown CSI between the source and sensors and unknown CSI between sensors and the GW under different configurations of the number of malicious sensors $\Pr(\mathcal{H}_B^s) \in \{0, 0.1, \dots, 0.6\}$, $(M = 100, L = 1)$

to.

6.3.1.2 Case II-Unknown CSI between the source and sensors and Unknown CSI between sensors and the GW

Fig. 6.5 presents ROC results for the case of unknown CSI between the source and sensors and unknown CSI between sensors and the GW (Section 6.1.4) for the optimal detection algorithm. Similarly as with Case I, the result shows that with the increase of the number of malicious users, the performance of the algorithm decreases. One observation is that Case II performs worse than Case I. This is expected as unknown channels between sensors and the GW add uncertainty to the detection problem.

Fig. 6.6 presents ROC results for case II as the mean $\bar{F}$ of the unknown channel between the source and sensors varies. Similarly as with Case I, the performance of the algorithm improves with increasingly $\bar{F}$.

6.3.1.3 Comparison with local decision based schemes

We compare our global decision algorithm (GD) with two other local decision based schemes (LD-1, LD-2). The result for Case I is shown in Figure 6.7. In LD-1, sensors make a local

Figure 6.6: Case II - ROC performance under unknown CSI between the source and sensors and unknown CSI between sensors and the GW under different configurations of $\bar{\mathbf{F}} \in \{0.1 + 0.1i, \dots, 0.45 + 0.45i\}$, ($M = 100, \Pr(\mathcal{H}_B^s) = 0.3, L = 1$)

Figure 6.7: Case I - ROC performance under unknown CSI between the source and sensors and known CSI between sensors and the GW for different schemes $\bar{\mathbf{F}} = 0.2 + 0.2i$, ($M = 100, \Pr(\mathcal{H}_B^s) = 0.3, L = 1$)

decision based on the information they receive and assume that a Byzantine attack function exists. Then each sensor transmits 0 or 1 to GW, where 1 represents $\mathcal{H}_1^\theta$ while 0 represents $\mathcal{H}_0^\theta$. According to the binary decisions by all the sensors, GW makes a final decision according to

majority rule, which means that if more than half of the sensors are shown to have observed the event, then the GW will decide that the event occurred. In LD-2, sensors make a binary decision without assuming the existence of a Byzantine attack function, then a malicious user may try to send opposite binary decisions from each sensor to the GW. The GW also makes a binary decision according to the information it receives assuming that a malicious function exists and by performing a majority rule.

The result shows that LD-1 and LD-2 perform worse compared with our scheme. This is expected because in our scheme, the GW makes a global optimal decision by taking into consideration information from all the sensors. LD-1 and LD-2 make local decisions for each sensor; therefore, a sub-optimal solution is achieved.

6.3.2 Distributed Event Detection Algorithm Nonlinear Transmission

Figure 6.8: ROC curve with respect to different number of sensors $M \in \{25, 35, 45\}$. Other parameters: $p_l = 0.5$, $P_{max} = 0.1$, $\sigma_w = \sigma_v = 0.1$, $\sigma_H = \sigma_G = 0.1$, $\mu_H = \mu_G = 0.2$, $K = 1$, $\theta = 1$

In this section, we present the detection performance of our algorithm in section 6.2 via Monte Carlo simulations. The setting for all the simulations are as follows: the prior distribution for all the channels is Rayleigh fading and the channels are assumed to be both spatially and temporally independent. We set the observed binary parameter, θ to 1 and the results are obtained from simulations over 5000 realizations of channel and noise for a given set of $N, M, L, \sigma_V, \sigma_W, \sigma_G, \sigma_F$. We show the different configurations: number of sensors, $M = \{25, 35, 45\}$. $p_l = \{1, 0.8, 0.4\}$, number of samples, $L = 1$ and noise variance, $\sigma_V^2 = \sigma_W^2 = 0.25$, $P_{min} = -P_{max} = \{-1, -0.8, -0.6\}$.

Fig. 6.8 presents ROC results for various values of number of sensors $M = \{25, 35, 45\}$. This result clearly shows the effect M has on the ROC performance. When number of sensors increases, the performance of the algorithm improves. When N increases, the more observations GW will have, which makes it easier for GW to tell which hypothesis the event belongs to.

Figure 6.9: ROC curve with respect to different proportion of linear vs. nonlinear sensors $p_l \in \{1, 0.8, 0.4\}$. $M = 30$, $P_{max} = 0.1$, $\sigma_w = \sigma_v = 0.1$, $\sigma_H = \sigma_G = 0.1$, $\mu_H = \mu_G = 0.2$, $K = 1$, $\theta = 1$

Fig. 6.9 presents ROC performance as a function of the p_l while M is 30. The result shows that with the increase of p_l , the performance of the algorithm decreases. This is expected as the higher p_l is, the more nonlinearity of observation at GW, which makes it more difficult for GW to distinguish between both hypotheses.

Figure 6.10: ROC curve with respect to different $P_{max} \in \{1, 0.8, 0.6\}$. $p_l = 0.5$. $M = 30$, $\sigma_w = \sigma_v = 0.1$, $\sigma_H = \sigma_G = 0.1$, $\mu_H = \mu_G = 0.2$

Fig. 6.10 presents ROC results as P_{min} and P_{max} vary. As expected, when the other parameters are fixed, the performance of algorithm improves with increasingly P_{max} . This is because as P_{max} increases, the signals are less nonlinear, which makes it easier for GW to make a correct decision.

6.4 Conclusion

In this chapter, we presented distributed event detection algorithms that consider WSNs with incomplete information from sensors, including sensors with byzantine attacks and nonlinear transmission.

We developed optimal Byzantine attack functions and optimal decision rules to decide whether the event occurred. We studied the optimal decision rules for two cases. We developed the optimal event detection decision rule under Byzantine attack for the first case and developed a novel low-complexity detection algorithm based on Gaussian approximation and moment matching for the second case. Through extensive simulations, we demonstrated the performance of the optimal decision rules under various scenarios. Our schemes outperformed two sub optimal algorithms which are based on local decisions. With extension of nonlinear transmission, we use distributed event detection algorithm with beta-jacobi series expansion. We also show the performance of algorithm using extensive simulations.

Chapter 7

Summary and Future Work

This chapter presents conclusions of the thesis and provides future works. In Section 7.1, the conclusion of the entire thesis is provided. In Section 7.2, we provide possible directions for future works.

7.1 Summary

In this thesis, we studied two important aspects of wireless sensors networks: firstly we designed various centralized and distributed clustering algorithms for maximizing network lifetime. Secondly, we studied the problem of event detection, aiming at deriving optimal decision rules in two types of deployment, namely, random WSNs and WSNs with incomplete information. We considered each problem separately and developed solutions for each of them (including algorithms, theoretical analysis, simulations). Along with the solutions, the theoretical analysis and extensive simulations prove the efficiency of proposed solutions.

We derive those algorithms starting from practical problems in WSNs. In particular, the centralized clustering algorithms can be used as a guideline for pre-deployment planning of WSN before the sensors are actually deployed. For example, the designers may want to know how many sensors need to be deployed and where they should be deployed in order to have a network with a certain lifetime. On the other hand, the proposed distributed algorithms are more suitable for practical deployment of large scale networks where sensors may have limited computational ability, even though we assume ideal conditions for the wireless channels, MAC protocol. For the event detection algorithm, the practical applications may be in participatory sensing or battlefield surveillance, as mentioned in the introduction. The designers may want

to know the deployment intensity required to satisfy an event detection threshold especially when the locations of the sensors are unknown to BS, which may result in a random network.

In Chapter 3, we proposed efficient (polynomial-time) EH node placement and clustering algorithms to maximize network lifetime, where the network lifetime is the duration until a certain proportion of the nodes run out of energy. Theoretical analysis and extensive simulations validated the near optimality of the proposed algorithms and demonstrated the benefits that can be gained by using EH sensors in order to prolong network lifetime in different scenarios.

In Chapter 4, we proposed an optimal distributed algorithm for scheduling a single battery powered relay node to serve the cluster heads to maximize network lifetime. We also proposed efficient distributed matching algorithm between CHs and multiple EH relay nodes to maximize network lifetime. Theoretical analysis and simulation results demonstrate much better performance compared with other approaches.

We then focused our attention on event detection algorithms in random WSNs. In Chapter 5, a binary hypothesis testing problem was formulated and optimal decision rule was designed for it. The marginal densities under two hypothesis were derived. We used simple yet accurate series expansion methods to approximate the marginal density under alternative hypothesis. The various series expansion methods were practical and suitable for different practical scenarios. We also used extensive simulation results to generate the Receiver Operating Curves (ROC) under different sets of parameters. The optimality of the series expansion methods was verified and the Skew-Kurtosis Curves were shown for each method.

In addition to the random WSNs, sensors in WSNs may send incomplete information to GW, either by Byzantine attackers or from nonlinear transmission. In Chapter 6, we presented distributed event detection algorithms that consider the case with incomplete information from sensors. We developed optimal decision rules to decide whether the event occurred. Through extensive simulations, we demonstrated the performance of our decision rules under various scenarios.

7.2 Future Work

There are many interesting and important extensions to the work developed in this thesis. They include:

- Study other configurations of introducing the EH nodes to the network: in this thesis, we considered EH nodes to serve as relay for CHs and single hop transmission in each cluster. It will be interesting to study EH nodes to serve as CHs or NCHs and multiple hop transmissions in each cluster.
- Extend our simulations to more realistic models, and implement and evaluate our algorithms in an actual WSN testbed: in this thesis, we used simulations in C/Matlab to show the performance of our algorithms and we assumed the EH node had constant energy harvesting rates. It will be interesting to evaluate our algorithms through actual testbed when the energy harvesting rates fluctuate frequently.
- Solve other event detection problem with multivariate series, multi-modal data and source estimation in WSNs. In this thesis, we studied event detection problem with univariate time series and single-modal data. It will be interesting to study multivariate time series data and multi-modal data.

In particular, future work for lifetime maximization and event detection problems include:

- (i) We shall consider the case where EH nodes are not necessarily serving as relay nodes between CHs and BS, but playing some other roles instead, e.g., a temporary CH until it has to be recharged again. The challenges of the problem is that CHs need to sense from the environment while collecting sensor data from NCHs. Once EH nodes run out of energy, NCHs have to find new CHs to join and once EH nodes become alive, it has to form new clusters again.
- (ii) Using real-life EH recharging profiles, we shall develop more realistic, more adaptive schemes to maximize the benefits of having EH nodes in clustered WSNs. We need to propose a stochastic auto-regressive model to model the real-life energy harvesting rates. More specifically, an alternative way is to measure the historic energy harvesting rate to validate the theoretical model.
- (iii) Other topics we are interested to study in the future include best allocations of EH nodes without a predefined number of clusters and EH clustering with enhanced robustness against unpredictable changes in weather, etc. The difficulty of this problem lies in finding the best number of CHs as well as designing adaptive schemes to changing weather conditions.

- (iv) It would be interesting to extend our results to the case where the transmitted power is unknown. We could then derive the Generalised Likelihood Ratio Test (GLRT) for this case. The challenges will be the difficulty of calculating the marginal likelihood since there will be two unknown random parameters including transmission power and distance.
- (v) A more general solution to the problem presented in Chapter 6 (saturated model) would be to consider a multivariate version of the Series Expansions instead of the univariate one. This can help improve the accuracy of the estimated marginal likelihood and lead to improved performance. For example, it is challenging to propose multivariate series expansions to capture the dependence of the channels.
- (vi) It would be interesting to extend the results to multiple hypothesis cases and to perform classification. For example, we could devise an algorithm to distinguish between different acoustic sources, such as shouting, gun shot and other sounds. The challenge is to set multiple thresholds and derive the test statistics for multiple hypotheses. Meanwhile, it is difficult to propose the model itself.

Appendix A

Appendix

A.1 Proof of Theorem 5.1

Proof. We begin by calculating p , as defined in Definition 5.1. According to [68], assume that $b_d(x, r)$ is contained within W and W is circle with Radius R . We then obtain that

$$p = \frac{\int_{b_d(x, r) \cap W} \lambda(x) dx}{\int_W \lambda(x) dx} = \frac{\int_0^r x^{-\nu} 2\pi x dx}{\int_0^R x^{-\nu} 2\pi x dx} = \begin{cases} \left(\frac{r}{R}\right)^{2-\nu}, & \nu < 2 \\ \frac{\log r - \log \varepsilon}{\log R - \log \varepsilon}, & \nu = 2 \\ \frac{r^{2-\nu} - \varepsilon^{2-\nu}}{R^{2-\nu} - \varepsilon^{2-\nu}}, & \nu > 2, \end{cases}$$

where $\varepsilon \rightarrow 0^+$.

Next, utilizing the results in [68], we can find the general expression for distribution distribution under inhomogeneous deployment. The procedure is same except that we use the expression for p from the above equation. For BPP, N_B is finite number. For IPPP, N is asymptotically $+\infty$. we obtain the following:

BPP deployment:

$$\begin{aligned}
 f_{R_k}(r|\mathbf{x}_s, \mathcal{H}_1) &= \frac{dp}{dr} \frac{(1-p)^{N_B-k} p^{k-1}}{B(N_B-k+1, k)} \\
 &= \frac{2-\nu}{R} \frac{(1-p)^{N_B-k} p^{k-1+\frac{1-\nu}{2-\nu}}}{B(N_B-k+1, k)} \\
 &= \frac{2-\nu}{R} \frac{B(k+\frac{1-\nu}{2-\nu}, N_B-k+1)}{B(N_B-k+1, k)} \beta\left(\left(\frac{r}{R}\right)^{2-\nu}; k+\frac{1-\nu}{2-\nu}, N_B-k+1\right) \\
 &= \frac{2-\nu}{R} \frac{\Gamma(k+\frac{1-\nu}{2-\nu}) \Gamma(N_B+1)}{\Gamma(k) \Gamma(N_B+\frac{1-\nu}{2-\nu}+1)} \beta\left(\left(\frac{r}{R}\right)^{2-\nu}; k+\frac{1-\nu}{2-\nu}, N_B-k+1\right).
 \end{aligned}$$

IPPP deployment:

$$\begin{aligned}
 f_{R_k}(r|\mathbf{x}_s, N=N_P, \mathcal{H}_1) &= \frac{2-\nu}{R} \frac{(1-p)^{N-k} p^{k-1+\frac{1-\nu}{2-\nu}}}{B(N-k+1, k)} \\
 &= \frac{2-\nu}{R} \frac{(1-p)^{N-k} p^{k-1+\frac{1-\nu}{2-\nu}} \Gamma(N+1)}{\Gamma(N-k+1) \Gamma(k)}.
 \end{aligned}$$

□

A.2 Proof of Lemma 5.1

Proof. We utilize the results for transformation of random variables to obtain:

BPP deployment:

$$\begin{aligned}
 f_{Z_k}(z) &= -f(\phi^{-1}(z)) \frac{d\phi^{-1}(z)}{dz} \\
 &= -C \beta\left(\left(\frac{z^{-2/\alpha}}{R}\right)^{2-\nu}; k+\frac{1-\nu}{2-\nu}, N_B-k+1\right) \left(-\frac{2}{\alpha}\right) z^{-2/\alpha-1} \\
 &= \frac{2(2-\nu)}{R\alpha} \frac{\Gamma(k+\frac{1-\nu}{2-\nu}) \Gamma(N_B+1)}{\Gamma(k) \Gamma(N_B+\frac{1-\nu}{2-\nu}+1)} \beta\left(\left(\frac{z^{-2/\alpha}}{R}\right)^{2-\nu}; k+\frac{1-\nu}{2-\nu}, N_B-k+1\right) z^{-2/\alpha-1}
 \end{aligned}$$

IPPP deployment:

Let $z = \phi(r) = r^{-\alpha/2}$, $y \in (0, \infty)$. Then

$$\begin{aligned} f_{Z_k}(z) &= -f_{R_k}(\phi^{-1}(z)) \frac{d\phi^{-1}(z)}{dz} \\ &= -f(z^{-2/\alpha}) (-2/\alpha) z^{-2/\alpha-1} \\ &= -\frac{(2\pi)^k}{\Gamma(k)(2-\nu)^{k-1}} z^{-2/\alpha((2-\nu)k-1)} \exp\left(\frac{2\pi}{2-\nu} z^{-2/\alpha(2-\nu)}\right) \left(-\frac{2}{\alpha}\right) z^{-2/\alpha-1} \\ &= \frac{(2\pi)^k}{\Gamma(k)(2-\nu)^{k-1}} \left(\frac{2}{\alpha}\right) z^{-2/\alpha((2-\nu)k)-1} \exp\left(\frac{2\pi}{2-\nu} z^{-2/\alpha(2-\nu)}\right). \end{aligned}$$

□

A.3 Proof of Theorem 5.2

Proof. To obtain the coefficients a_i , we need to solve the following expression:

$$\begin{aligned} a_i &= \frac{B(\theta, \eta)(2i + \theta + \eta - 1)\Gamma(i + \theta + \eta - 1)i!}{\Gamma(i + \theta)\Gamma(i + \eta)} \int_{-1}^1 f(x) P_i^{(\eta-1, \theta-1)}(x) dx \\ &= \frac{B(\theta, \eta)(2i + \theta + \eta - 1)i!}{\Gamma(i + \theta)} \sum_{m=0}^i \frac{\Gamma(\theta + \eta + i + m - 1)}{\Gamma(i - m + 1)\Gamma(\eta + m)m!} \int_{-1}^1 f(x) \left(\frac{x-1}{2}\right)^m dx. \end{aligned}$$

Using the following series expansion:

$$\left(\frac{x-1}{2}\right)^m = \frac{1}{2^m} \sum_{j=0}^m \binom{m}{j} (-1)^{m-j} x^j dx,$$

and after some algebraic manipulations, we obtain that

$$a_i = \sum_{j=0}^i \mu_j \frac{B(\theta, \eta)(2i + \theta + \eta - 1)i!}{\Gamma(i + \theta)} \sum_{m=j}^i \frac{\Gamma(\theta + \eta + i + m - 1)}{\Gamma(i - m + 1)\Gamma(\eta + m)m!2^m} \binom{m}{j} (-1)^{m-j}.$$

where $\mu_j = \mathbb{E}[X^j] = \int_{-1}^1 f(x) x^j dx$.

□

A.4 Proof of Theorem 5.3

Proof. We express the fourth order Beta-Jacobi series expansion as follows:

$$f(x) = \frac{(x+1)^{\theta-1}(1-x)^{\eta-1}}{B(\theta, \eta)2^{\theta+\eta-1}} \sum_{i=0}^4 a_i P_i^{(\eta-1, \theta-1)}(x) = B_1 \mu_3 + B_2 \mu_4 + B_3,$$

where

$$\begin{aligned}
 B_1 &= (C_{33}P_3 + C_{43}P_4) \frac{(x+1)^{\theta-1} (1-x)^{\eta-1}}{B(\theta, \eta) 2^{\theta+\eta-1}}, \\
 B_2 &= C_{44}P_4 \frac{(x+1)^{\theta-1} (1-x)^{\eta-1}}{B(\theta, \eta) 2^{\theta+\eta-1}}, \\
 B_3 &= \frac{(x+1)^{\theta-1} (1-x)^{\eta-1}}{B(\theta, \eta) 2^{\theta+\eta-1}} \left(\sum_{i=0}^2 a_i P_i + (C_{30} + C_{31}\mu_1 + C_{32}\mu_2) P_3 + (C_{40} + C_{41}\mu_1 + C_{42}\mu_2) P_4 \right), \\
 C_{ij} &= \frac{B(\theta, \eta) (2i + \theta + \eta - 1) i!}{\Gamma(i + \theta)} \sum_{m=j}^i \frac{\Gamma(\eta + \theta + i + m - 1)}{\Gamma(i - m + 1) \Gamma(\eta + m) m! 2^m} \binom{m}{j} (-1)^{m-j},
 \end{aligned}$$

and $\mu_1 = \mathbb{E}[X]$, $\mu_2 = \mathbb{E}[X^2]$.

In order to guarantee that $f(x)$ is positive, we obtain the following set of equations

$$\begin{cases} B_1\mu_3 + B_2\mu_4 + B_3 = 0, & (f(x) = 0) \\ B'_1\mu_3 + B'_2\mu_4 + B'_3 = 0, & (f'(x) = 0) \end{cases} \quad \text{for } x \in [-1, 1]$$

Solving for the Skew and Kurtosis, we obtain that:

$$\begin{cases} \mu_3(x) &= -\frac{1}{B_1} (\mu_4(x) B_2 + B_3) \\ \mu_4(x) &= \left(\frac{B'_1 B_3}{B_1} - B'_3 \right) \left(B'_2 - \frac{B'_1 B_2}{B_1} \right)^{-1}, \end{cases} \quad \text{for } x \in [-1, +1)$$

where

$$\begin{aligned}
 B'_1 &= (C_{33}P'_3 + C_{43}P'_4) \frac{(x+1)^{\theta-1} (1-x)^{\eta-1}}{B(\theta, \eta) 2^{\theta+\eta-1}} + B_1 \left(\frac{\theta+1}{x+1} - \frac{\eta-1}{1-x} \right), \\
 B'_2 &= C_{44}P'_4 \frac{(x+1)^{\theta-1} (1-x)^{\eta-1}}{B(\theta, \eta) 2^{\theta+\eta-1}} + B_2 \left(\frac{\theta+1}{x+1} - \frac{\eta-1}{1-x} \right), \\
 B'_3 &= \left(\sum_{i=0}^2 a_i P'_i + (C_3 + C_{31}\mu_1 + C_{32}\mu_2) P'_3 + (C_4 + C_{41}\mu_1 + C_{42}\mu_2) P'_4 \right) \frac{(x+1)^{\theta-1} (1-x)^{\eta-1}}{B(\theta, \eta) 2^{\theta+\eta-1}} \\
 &\quad + B_3 \left(\frac{\theta+1}{x+1} - \frac{\eta-1}{1-x} \right),
 \end{aligned}$$

and

$$P'_0 = 0$$

$$P'_i = \frac{\Gamma(\eta + i)}{\Gamma(\eta + \theta + i - 1)} \left(\sum_{m=1}^i \frac{\Gamma(\eta + \theta + i + m - 1)}{2\Gamma(i - m + 1)\Gamma(\eta + m)(m-1)!} \left(\frac{x-1}{2} \right)^{m-1} \right).$$

□

A.5 Proof of Theorem 5.4

Proof. (i) BPP deployment:

We define the following change of variables: $X = \left(\frac{Z^{-2/\alpha}}{R}\right)^{2-\nu}$. We now express the m -th moment of Z_k as:

$$\begin{aligned}
 \mathbb{E}[Z_k^m] &= \int w(k) \beta\left(x; k + \frac{1-\nu}{2-\nu}, N_B - k + 1\right) \left(x^{-\frac{\alpha}{2(2-\nu)}} R^{-\alpha/2}\right)^{-\frac{2}{\alpha}+m-1} \frac{dx^{-\frac{\alpha}{2(2-\nu)}} R^{-\alpha/2}}{dx} dx \\
 &= C_1 \int \beta\left(x; k + \frac{1-\nu}{2-\nu}, N_B - k + 1\right) x^{\frac{1}{2-\nu} + \frac{\alpha}{2(2-\nu)} - \frac{m\alpha}{2(2-\nu)}} x^{-\frac{\alpha}{2(2-\nu)}-1} dx \\
 &= C_1 \int_0^1 \beta\left(x; k + \frac{1-\nu}{2-\nu}, N_B - k + 1\right) x^{\frac{1}{2-\nu} - \frac{m\alpha}{2(2-\nu)}-1} dx \\
 &= C_1 \int_0^1 \frac{1}{B\left(k + \frac{1-\nu}{2-\nu}, N_B - k + 1\right)} x^{k + \frac{1-\nu}{2-\nu} - 1 + \frac{1}{2-\nu} - \frac{m\alpha}{2(2-\nu)}-1} (1-x)^{N_B-k} dx \\
 &= C_1 \int_0^1 \frac{1}{B\left(k + \frac{1-\nu}{2-\nu}, N_B - k + 1\right)} x^{k - \frac{m\alpha}{2(2-\nu)}-1} (1-x)^{N_B-k} dx \\
 &= C_1 \frac{B\left(k - \frac{m\alpha}{2(2-\nu)}, N_B - k + 1\right)}{B\left(k + \frac{1-\nu}{2-\nu}, N_B - k + 1\right)},
 \end{aligned}$$

where $B(\cdot)$ is the β function and

$$\begin{aligned}
 C_1 &= w(k) (R^{-\alpha/2})^{-\frac{2}{\alpha}-1+m+1} \left(-\frac{\alpha}{2(2-\nu)}\right) \\
 &= \frac{2-\nu}{R} \frac{\Gamma\left(k + \frac{1-\nu}{2-\nu}\right) \Gamma(N_B + 1)}{\Gamma(k) \Gamma\left(N_B + \frac{1-\nu}{2-\nu} + 1\right)} \left(-\frac{2}{\alpha}\right) (R^{-\alpha/2})^{-\frac{2}{\alpha}+m} \left(-\frac{\alpha}{2(2-\nu)}\right) \\
 &= R^{-m\alpha/2} \frac{\Gamma\left(k + \frac{1-\nu}{2-\nu}\right) \Gamma(N_B + 1)}{\Gamma(k) \Gamma\left(N_B + \frac{1-\nu}{2-\nu} + 1\right)}.
 \end{aligned}$$

Finally we obtain that

$$\begin{aligned}
 \mathbb{E}[Z_{k_i}^m] &= R^{-m\alpha/2} \frac{\Gamma\left(k + \frac{1-\nu}{2-\nu}\right) \Gamma(N_B + 1)}{\Gamma(k) \Gamma\left(N_B + \frac{1-\nu}{2-\nu} + 1\right)} \frac{B\left(k - \frac{m\alpha}{2(2-\nu)}, N_B - k + 1\right)}{B\left(k + \frac{1-\nu}{2-\nu}, N_B - k + 1\right)} \\
 &= R^{-m\alpha/2} \frac{\Gamma(N_B + 1) \Gamma\left(k - \frac{m\alpha}{2(2-\nu)}\right)}{\Gamma(k) \Gamma\left(N_B - \frac{m\alpha}{2(2-\nu)} + 1\right)}.
 \end{aligned}$$

(ii) IPPP deployment: We express the m -th moment as follows:

$$\begin{aligned}\mathbb{E}[Z_k^m] &= \int_0^\infty z^m f_{Z_k}(z) dz \\ &= \int_0^\infty z^m \frac{(2\pi)^k}{\Gamma(k)(2-\nu)^{k-1}} \left(\frac{2}{\alpha}\right) z^{-2/\alpha((2-\nu)k)-1} \exp\left(-\frac{2\pi}{c+2} z^{-2/\alpha(c+2)}\right) dz \\ &= \frac{2}{\alpha} \frac{(2\pi)^k}{\Gamma(k)(2-\nu)^{k-1}} \int_0^\infty z^{m-2(2-\nu)k/\alpha-1} \exp\left(-\frac{2\pi}{2-\nu} z^{-2/\alpha(c+2)}\right) dz.\end{aligned}$$

Using the identity $\Gamma(t) = \int_0^\infty x^{t-1} \exp(-x) dx$, and using the following change of variables $X = \frac{2\pi}{c+2} Z^{-2/\alpha(2-\nu)}$, we obtain that

$$\begin{aligned}\mathbb{E}[Z_k^m] &= \frac{2}{\alpha} \frac{(2\pi)^k}{\Gamma(k)(2-\nu)^{k-1}} \int_0^\infty \left(\frac{x(2-\nu)}{2\pi}\right)^{-\frac{\alpha}{2(2-\nu)}(m-2(2-\nu)k/\alpha-1)} \exp(-x) \frac{d\left(\frac{x(2-\nu)}{2\pi}\right)^{-\frac{\alpha}{2(2-\nu)}}}{dx} dx \\ &= \frac{2}{\alpha} \frac{(2\pi)^k}{\Gamma(k)(2-\nu)^{k-1}} \frac{\alpha}{2(2-\nu)} \left(\frac{2-\nu}{2\pi}\right)^{-\frac{\alpha m}{2(2-\nu)}+k} \int_0^\infty \exp(-x) x^{k-\frac{\alpha m}{2(2-\nu)}-1} dx \\ &= \left(\frac{2-\nu}{2\pi}\right)^{-\frac{\alpha m}{2(2-\nu)}} \frac{\Gamma\left(k - \frac{\alpha m}{2(2-\nu)}\right)}{\Gamma(k)}.\end{aligned}$$

□

A.6 Proof of Lemma 5.5

Proof. (i) BPP deployments:

Using the result in Theorem 5.4, MGF of Z_k is given by:

$$M_{Z_k}(t) = 1 + \sum_{m=1}^{\infty} \frac{t^m}{m!} R^{-m\alpha/2} \frac{\Gamma(N_B + 1) \Gamma\left(k - \frac{m\alpha}{2(2-\nu)}\right)}{\Gamma(k) \Gamma\left(N_B - \frac{m\alpha}{2(2-\nu)} + 1\right)}$$

Due to the conditional independence of Z_k , and that the MGF of W_l is given by $M_{W_l}(t) = \exp(\frac{1}{2}\sigma_{w_l}^2 t^2)$, we have that

$$\begin{aligned}M_{Y_l}(t) &= \prod_{k=1}^{N_B} M_{Z_k}(t) M_{W_l}(t) \\ &= \prod_{k=1}^{N_B} \left(1 + \sum_{m=1}^{\infty} \frac{(\sqrt{P_0}t)^m}{m!} R^{-m\alpha/2} \frac{\Gamma(N_B + 1) \Gamma\left(k - \frac{m\alpha}{2(2-\nu)}\right)}{\Gamma(k) \Gamma\left(N_B - \frac{m\alpha}{2(2-\nu)} + 1\right)}\right) e^{\frac{1}{2}\sigma_w^2 t^2}.\end{aligned}$$

(ii) IPPP deployments:

Using the result in Theorem 5.4, MGF of Z_k is given by:

$$M_{Z_k}(t) = 1 + \sum_{m=1}^{\infty} \frac{t^m}{m!} \left(\frac{2-\nu}{2\pi} \right)^{-\frac{\alpha m}{2(2-\nu)}} \frac{\Gamma\left(k - \frac{\alpha m}{2(2-\nu)}\right)}{\Gamma(k)}$$

Using the same procedure as before, we obtain that

$$M_{Y_i}(t) = \prod_{k=1}^{N_P} \left(1 + \sum_{m=1}^{\infty} \frac{(\sqrt{P_0}t)^m}{m!} \left(\frac{2-\nu}{2\pi} \right)^{-\frac{\alpha m}{2(2-\nu)}} \frac{\Gamma\left(k - \frac{\alpha m}{2(2-\nu)}\right)}{\Gamma(k)} \right) \sigma_W^2 t^2.$$

In order to derive cumulants and moments, first we find the cumulant function $g(t)$ under both deployments.

(i) BPP deployment:

$$\begin{aligned} g(t) &= \log M_{Y_i}(t) \\ &= \sum_{k=1}^{N_B} \log \left(1 + \sum_{m=1}^{\infty} \frac{(\sqrt{P_0}t)^m}{m!} R^{-\frac{\alpha}{2}m} h(k) \right) + \frac{1}{2} \sigma_w^2 t^2, \end{aligned}$$

where

$$h(k) := R^{-\alpha/2} \frac{\Gamma(N_B + 1) \Gamma\left(k - \frac{m\alpha}{2(2-\nu)}\right)}{\Gamma(k) \Gamma\left(N_B - \frac{m\alpha}{2(2-\nu)} + 1\right)}.$$

Next we obtain the first cumulant κ_1 as follows:

$$\begin{aligned} \kappa_1 &= \left. \frac{dg(t)}{dt} \right|_{t=0} \\ &= \sum_{k=1}^{N_B} \frac{\frac{\sqrt{P_0}t^0}{1} h(k) + \sum_{m=2}^{\infty} m \frac{(\sqrt{P_0})^m t^{m-1}}{m!} h(k)}{1 + \sum_{m=1}^{\infty} \frac{(\sqrt{P_0}t)^m}{m!} h(k)} \Big|_{t=0} + \sigma_W^2 t \Big|_{t=0} \\ &= \sum_{k=1}^{N_B} \sqrt{P_0} h(k) \\ &= \sum_{k=1}^{N_B} \sqrt{P_0} \mathbb{E}[Z_k]. \end{aligned}$$

Similarly, by taking the second, third and forth derivative of $g(t)$. We next find the κ_2 , κ_3 and κ_4 .

$$\begin{aligned}\kappa_2 &= \sum_{k=1}^{N_B} P_0 \left(\mathbb{E} [Z_k^2] - (\mathbb{E} [Z_k])^2 \right) + \sigma_W^2, \\ \kappa_3 &= \sum_{k=1}^{N_B} \sqrt{P_0}^3 \left(\mathbb{E} [Z_k^3] - 3\mathbb{E} [Z_k^2] \mathbb{E} [Z_k] + 2(\mathbb{E} [Z_k])^3 \right), \\ \kappa_4 &= \sum_{k=1}^{N_B} P_0^2 \left(\mathbb{E} [Z_k^4] - 4\mathbb{E} [Z_k^3] \mathbb{E} [Z_k] - 3(\mathbb{E} [Z_k])^2 + 12\mathbb{E} [Z_k^2] (\mathbb{E} [Z_k])^2 - 6(\mathbb{E} [Z_k])^4 \right).\end{aligned}$$

(ii) IPPP deployment:

κ_1 , κ_2 , κ_3 and κ_4 are in the same form as those in BPP deployment using the appropriate values for the moments.

□

A.7 Proof of Theorem 6.1

Proof. The distribution of $\mathcal{F}_L(R_m(l) | \mathcal{H}_{j=0,1})$ is given by:

$$\mathcal{F}_L(R_m(l) | \mathcal{H}_{j=0,1}) \sim N(\mu_j, \sigma_j),$$

where $\mu_0 = 0$, $\mu_1 = g\bar{F}\theta$, $\sigma_0^2 = g^2\sigma_V^2$ and $\sigma_1^2 = g^2\theta^2\sigma_V^2 + g^2\sigma_H^2$.

Its k -th moment is given by:

$$\mathbb{E} \left[\mathcal{F}_L(R_m(l))^k | \mathcal{H}_{j=0,1} \right] = \sigma_j^k \left(-i\sqrt{2}\text{sgn}(\mu_j)^k \right) U \left(-\frac{1}{2}k, \frac{1}{2}, -\frac{1}{2}(\mu_j/\sigma_j)^2 \right).$$

Next, we calculate $\mathbb{E} \left[\mathcal{F}_{NL}(R_m(l))^k | \mathcal{H}_{j=0,1} \right]$.

$$\begin{aligned}\mathbb{E} \left[\mathcal{F}_{NL}(R_m(l))^k | \mathcal{H}_{j=1,2} \right] &= \\ &= \int_{-\infty}^{P_{\min}} (KP_{\min})^k \phi \left(\frac{x - \mu_j}{\sigma_j} \right) \mathfrak{x} + \int_{P_{\min}}^{P_{\max}} (Kx)^k \phi \left(\frac{x - \mu_j}{\sigma_j} \right) \mathfrak{x} + \int_{P_{\max}}^{\infty} (KP_{\max})^k \phi \left(\frac{x - \mu_j}{\sigma_j} \right) \mathfrak{x} \\ &= (KP_{\min})^k \Phi \left(\frac{P_{\min} - \mu_j}{\sigma_j} \right) + \int_{P_{\min}}^{P_{\max}} (Kx)^k \phi \left(\frac{x - \mu_j}{\sigma_j} \right) \mathfrak{x} + (KP_{\max})^k \left(1 - \Phi \left(\frac{P_{\max} - \mu_j}{\sigma_j} \right) \right)\end{aligned}$$

where $\mu_0 = 0$, $\sigma_0 = \sigma_v$, $\mu_1 = \bar{F}\theta$, $\sigma_1 = \sqrt{\theta^2 \bar{F} + \sigma_v^2}$.

Now we calculate $\int_{P_{\min}}^{P_{\max}} (Kx)^k \phi\left(\frac{x-\mu_j}{\sigma_j}\right) \mathfrak{x}$.

$$\int_{P_{\min}}^{P_{\max}} (Kx)^k \phi\left(\frac{x-\mu_j}{\sigma_j}\right) \mathfrak{x} = K^k \left(\int_{-\infty}^{P_{\max}} x^k \phi\left(\frac{x-\mu_j}{\sigma_j}\right) \mathfrak{x} - \int_{-\infty}^{P_{\min}} x^k \phi\left(\frac{x-\mu_j}{\sigma_j}\right) \mathfrak{x} \right)$$

Using a change of variable $a := \frac{x-\mu_j}{\sigma_j}$ and using the identity

$$(ax + b)^k = \sum_{r=0}^k \binom{k}{r} b^{k-r} a^r x^r,$$

the integrals have the following generic form:

$$\begin{aligned} \int_{-\infty}^c (ax + b)^k \phi(x) \mathfrak{x} &= \int_{-\infty}^c \sum_{r=0}^k \binom{k}{r} b^{k-r} a^r x^r \phi(x) \mathfrak{x} \\ &= \sum_{r=0}^k \binom{k}{r} b^{k-r} a^r \int_{-\infty}^c x^r \phi(x) \mathfrak{x} \\ &= b^k \Phi(c) + \sum_{r=1}^k \binom{k}{r} b^{k-r} a^r \int_{-\infty}^c x^r \phi(x) \mathfrak{x} \\ &= b^k \Phi(c) - \sum_{r \text{ is odd}} \binom{k}{r} b^{k-r} a^r \phi(c) \sum_{j=0}^{\frac{r-1}{2}} \frac{(r-1)!!}{(2j)!!} c^{2j} \\ &\quad - \sum_{r \text{ is even}} \left(\binom{k}{r} b^{k-r} a^r \phi(c) \sum_{j=0}^{\frac{r-2}{2}} \frac{(r-1)!!}{(2j+1)!!} c^{2j+1} + (r-1)!! \Phi(c) \right) \end{aligned} \tag{A.1}$$

where !! denotes the double factorial symbol, and $a = \sigma_j$, $b = \mu_j$. Finally we obtain:

$$\begin{aligned} \mathbb{E} \left[\mathcal{F}_{\text{NL}}(R_m(l))^k | \mathcal{H}_{j=1,2} \right] &= (K P_{\min})^k \Phi\left(\frac{P_{\min} - \mu_j}{\sigma_j}\right) + K^k (I_{P_{\max}} - I_{P_{\min}}) \\ &\quad + (K P_{\max})^k \left(1 - \Phi\left(\frac{P_{\max} - \mu_j}{\sigma_j}\right) \right). \end{aligned}$$

$I_{P_{\max}}$ and $I_{P_{\min}}$ can be derived from (A.1) by setting $c = \frac{P_{\max} - \mu_j}{\sigma_j}$ or $c = \frac{P_{\min} - \mu_j}{\sigma_j}$ respectively.

□

References

- [1] M. Petar, “A distributed switch scheduling algorithm,” *Performance Evaluation*, vol. 64, pp. 1053 – 1061, Oct 2007.
- [2] N. H. Mak and W. Seah, “How long is the lifetime of a wireless sensor network?,” *International Conference on Advanced Information Networking and Applications*, pp. 763 –770, may 2009.
- [3] D. Dechene, A. E. Jardali, M. Luccini, and A. Sauer, “A survey of clustering algorithms for wireless sensor networks,” *Computer Communications archive*, vol. 30, pp. 2826–2841, Oct 2007.
- [4] S. Sudevalayam and P. Kulkarni, “Energy harvesting sensor nodes: Survey and implications,” *IEEE Communications Surveys Tutorials*, vol. 13, pp. 443–461, Third Quarter 2011.
- [5] R. Vullers, R. Schaijk, H. Visser, J. Penders, and C. Hoof, “Energy harvesting for autonomous wireless sensor networks,” *IEEE Solid-State Circuits Magazine*, vol. 2, no. 2, pp. 29–38, 2010.
- [6] T. Arampatzis, J. Lygeros, and S. Manesis, “A survey of applications of wireless sensors and wireless sensor networks,” *Proceedings of the 2005 IEEE International Symposium on Intelligent Control*, pp. 719 –724, 2005.
- [7] T. Prabhakar, S. Devasenapathy, H. Jamadagni, and R. Prasad, “Smart applications for energy harvested wsns,” *Second International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1 –7, Jan 2010.

- [8] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393 – 422, 2002.
- [9] A. Rawat, P. Anand, H. Chen, and P. K. Varshney, "Collaborative spectrum sensing in the presence of byzantine attacks in cognitive radio networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 774–786, 2011.
- [10] B. Kailkhura, S. Brahma, Y. S. Han, and P. K. Varshney, "Optimal distributed detection in the presence of byzantines," pp. 2925–2929, 2013.
- [11] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. WILEY, Hoboken, NJ, 2005.
- [12] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826 – 2841, 2007.
- [13] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, vol. 2, p. 223, Jan. 2000.
- [14] M. Ye, C. Li, G. Chen, and J. Wu, "Eecs: an energy efficient clustering scheme in wireless sensor networks," *24th IEEE International Performance, Computing, and Communications Conference*, pp. 535 – 540, April 2005.
- [15] O. Younis and S. Fahmy, "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, pp. 366 – 379, Oct.-Dec. 2004.
- [16] M. Zhang, C. Gong, and Y. Lu, "An novel dynamic clustering algorithm based on geographical location for wireless sensor networks," *International Symposium on Information Science and Engineering*, vol. 1, pp. 565 –568, Dec. 2008.
- [17] Y.-F. Huang, W.-H. Luo, J. Sum, L.-H. Chang, C.-W. Chang, and R.-C. Chen, "Lifetime performance of an energy efficient clustering algorithm for cluster-based wireless sensor networks," *Frontiers of High Performance Computing and Networking ISPA 2007 Workshops*, vol. 4743, pp. 455–464, 2007.

REFERENCES

- [18] J. Peng, L. Li, and S. Xu, "A novel energy efficient and reliable clustering algorithm in wireless sensor networks," *IET Conference Publications*, vol. 2007, no. 533, pp. 596–599, 2007.
- [19] K. Xu and M. Gerla, "A heterogeneous routing protocol based on a new stable clustering scheme," *Proceedings - IEEE Military Communications Conference MILCOM*, vol. 2, pp. 838–843, 2002.
- [20] T.-C. Hou and T.-J. Tsai, "A access-based clustering protocol for multihop wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 1201–1210, Jul. 2001.
- [21] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," *IEEE INFOCOM*, vol. 3, pp. 1567 – 1576 vol.3, 2002.
- [22] T. Van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," *SenSys'03: Proceedings of the First International Conference on Embedded Networked Sensor Systems*, pp. 171–180, 2003.
- [23] A. Iranli, M. Maleki, and M. Pedram, "Energy efficient strategies for deployment of a two-level wireless sensor network," *Proceedings of the 2005 International Symposium on Low Power Electronics and Design*, pp. 233 – 238, Aug. 2005.
- [24] N. Aslam, W. Phillips, and W. Robertson, "Effective location management of mobile actors in wireless sensor and actor networks," *Journal of Networks*, vol. 4, no. 1, pp. 1–8, 2009.
- [25] X. Ning and C. G. Cassandras, "Optimal cluster-head deployment in wireless sensor networks with redundant link requirements," *Proceedings of the 2nd international conference on Performance evaluation methodologies and tools*, pp. 241–249, 2007.
- [26] A. Chehri and H. Mouftah, "An efficient clusterhead placement for hybrid sensor networks," *Ad-Hoc, Mobile and Wireless Networks*, vol. 6288, pp. 123–134, 2010.
- [27] K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An efficient clustering-based heuristic for data gathering and aggregation in sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1948–1953, 2003.

- [28] K. Dasgupta, M. Kukreja, and K. Kalpakis, "Topology-aware placement and role assignment for energy-efficient information gathering in sensor networks," *In IEEE ISCC*, pp. 341–348, 2003.
- [29] M. Younis, M. Youssef, and K. Arisha, "Energy-aware management for cluster-based sensor networks," *Computer Networks*, vol. 43, no. 5, pp. 649 – 668, 2003.
- [30] P. Ding, J. Holliday, and A. Celik, "Distributed energy-efficient hierarchical clustering for wireless sensor networks," *Distributed Computing in Sensor Systems*, vol. 3560, pp. 322–339, 2005.
- [31] L. Qing, Q. Zhu, and M. Wang, "Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks," *Computer Communications*, vol. 29, no. 12, pp. 2230 – 2237, 2006.
- [32] G. Smaragdakis, I. Matta, and A. Bestavros, "SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks," *Proc. of the Int'l Workshop on SANPA*, 2004.
- [33] S. Banerjee and S. Khuller, "A clustering scheme for hierarchical control in multi-hop wireless networks," *Proceedings - IEEE INFOCOM*, vol. 2, pp. 1028–1037, 2001.
- [34] E. Oyman and C. Ersoy, "Multiple sink network design problem in large scale wireless sensor networks," *IEEE International Conference on Communications*, vol. 6, pp. 3663–3667, 2004.
- [35] H. Gou, Y. Yoo, and H. Zeng, "A partition-based leach algorithm for wireless sensor networks," *Ninth IEEE International Conference on Computer and Information Technology*, vol. 2, pp. 40 –45, oct. 2009.
- [36] N. Aslam, S. Sivakumar, W. Phillips, and W. Robertson, "Energy efficient cluster formation using a multi-criterion optimization technique for wireless sensor networks," *2007 4th Annual IEEE Consumer Communications and Networking Conference, CCNC 2007*, pp. 650–654, 2007.

REFERENCES

- [37] A. Manjeshwar and D. Agrawal, “Teen: a routing protocol for enhanced efficiency in wireless sensor networks,” *Proceedings 15th International Parallel and Distributed Processing Symposium*, pp. 2009 –2015, apr 2001.
- [38] D. Kumar, T. C. Aseri, and R. Patel, “Eehc: Energy efficient heterogeneous clustered scheme for wireless sensor networks,” *Computer Communications*, vol. 32, no. 4, pp. 662 – 667, 2009.
- [39] O. Buyanjargal and Y. Kwon, “Aeec-adaptive and energy efficient clustering algorithm for content based wireless sensor networks,” *Computer Science and its Applications, 2009. CSA '09. 2nd International Conference on*, pp. 1 –6, dec. 2009.
- [40] M. Gorlatova, P. Kinget, I. Kymissis, D. Rubenstein, X. Wang, and G. Zussman, “Challenge: ultra-low-power energy-harvesting active networked tags (enhants),” *Proceedings of the 15th annual international conference on Mobile computing and networking*, pp. 253–260, 2009.
- [41] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, “Design considerations for solar energy harvesting wireless embedded systems,” *Fourth International Symposium on Information Processing in Sensor Networks*, pp. 457 – 462, April 2005.
- [42] D. Hasenfratz, A. Meier, C. Moser, J.-J. Chen, and L. Thiele, “Analysis, comparison, and optimization of routing protocols for energy harvesting wireless sensor networks,” *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*, pp. 19 –26, june 2010.
- [43] C. Bergonzini, D. Brunelli, and L. Benini, “Algorithms for harvested energy prediction in batteryless wireless sensor networks,” *3rd International Workshop on Advances in sensors and Interfaces*, pp. 144 –149, june 2009.
- [44] M. Islam, M. Islam, and M. Islam, “A-sleach: An advanced solar aware leach protocol for energy efficient routing in wireless sensor networks,” *Sixth International Conference on Networking*, p. 4, April 2007.

- [45] B. Medepally and N. Mehta, “Voluntary energy harvesting relays and selection in cooperative wireless networks,” *IEEE Transactions on Wireless Communications*, vol. 9, pp. 3543–3553, November 2010.
- [46] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: A survey,” *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.
- [47] F. Fazel, M. Fazel, and M. Stojanovic, “Random access sensor networks: Field reconstruction from incomplete data,” *IEEE Information Theory and Applications Workshop (ITA)*, pp. 300–305, 2012.
- [48] J. Matamoros, F. Fabbri, C. Antón-Haro, and D. Dardari, “On the estimation of randomly sampled 2d spatial fields under bandwidth constraints,” *IEEE Transactions on Wireless Communications*, vol. 10, no. 12, pp. 4184–4192, 2011.
- [49] R. Viswanathan and P. K. Varshney, “Distributed detection with multiple sensors i. fundamentals,” *Proceedings of the IEEE*, vol. 85, no. 1, pp. 54–63, 1997.
- [50] J.-J. Xiao and Z.-Q. Luo, “Universal decentralized detection in a bandwidth-constrained sensor network,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2617–2624, 2005.
- [51] T. Aysal and K. Barner, “Constrained decentralized estimation over noisy channels for sensor networks,” *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1398–1410, 2008.
- [52] M. Zhu, S. Ding, Q. Wu, R. R. Brooks, N. S. Rao, and S. S. Iyengar, “Fusion of threshold rules for target detection in wireless sensor networks,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, p. 18, 2010.
- [53] M. Guerriero, L. Svensson, and P. Willett, “Bayesian data fusion for distributed target detection in sensor networks,” *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3417–3421, 2010.

REFERENCES

- [54] J.-F. Chamberland and V. V. Veeravalli, “Wireless sensors in distributed detection applications,” *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 16–25, 2007.
- [55] P. Karumbu, V. K. Prasanthi, and A. Kumar, “Delay optimal event detection on ad hoc wireless sensor networks,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 2, p. 12, 2012.
- [56] D. Bajovic, B. Sinopoli, and J. Xavier, “Sensor selection for event detection in wireless sensor networks,” *Signal Processing, IEEE Transactions on*, vol. 59, no. 10, pp. 4938–4953, 2011.
- [57] D. Ciuonzo, G. Romano, and P. Rossi, “Performance analysis and design of maximum ratio combining in channel-aware mimo decision fusion,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4716–4728, 2013.
- [58] I. Nevat, G. W. Peters, and I. B. Collings, “Distributed detection in sensor networks over fading channels with multiple antennas at the fusion centre,” *accepted to IEEE Transactions on Signal Processing*, 2013.
- [59] X. Zhang, H. Poor, and M. Chiang, “Optimal power allocation for distributed detection over mimo channels in wireless sensor networks,” *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4124–4140, 2008.
- [60] P. Zhang, J. Y. Koh, S. Lin, and I. Nevat, “Distributed event detection under byzantine attack in wireless sensor networks,” in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pp. 1–6, IEEE, 2014.
- [61] Y. Taniguchi, T. Kitani, and K. Leibnitz, “A uniform airdrop deployment method for large-scale wireless sensor networks,” *International Journal of Sensor Networks*, vol. 9, no. 3, pp. 182–191, 2011.
- [62] F. Oldewurtel and P. Mahonen, “Analysis of enhanced deployment models for sensor networks,” in *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, pp. 1–5, IEEE, 2010.

REFERENCES

- [63] “Preserving privacy in participatory sensing systems,” *Computer Communications*, vol. 33, no. 11, pp. 1266 – 1280, 2010.
- [64] J. G. Andrews, F. Baccelli, and R. K. Ganti, “A tractable approach to coverage and rate in cellular networks,” *Communications, IEEE Transactions on*, vol. 59, no. 11, pp. 3122–3134, 2011.
- [65] J. G. Andrews, R. K. Ganti, M. Haenggi, N. Jindal, and S. Weber, “A primer on spatial modeling and analysis in wireless networks,” *Communications Magazine, IEEE*, vol. 48, no. 11, pp. 156–163, 2010.
- [66] Y. Cho, A. Galstyan, J. Brantingham, and G. Tita, “Generative models for spatial-temporal processes with applications to predictive criminology,” 2012.
- [67] M. C. Guenther and J. T. Bradley, “On performance of gossip communication in a crowd-sensing scenario,” *Quantitative Evaluation of Systems*, pp. 122–137, 2014.
- [68] M. Haenggi, “On distances in uniformly random networks,” *IEEE Transactions on Information Theory*, vol. 51, pp. 3584–3586, Oct 2005.
- [69] D. Moltchanov, “Distance distributions in random networks,” *Ad Hoc Networks*, vol. 10, no. 6, pp. 1146 – 1166, 2012.
- [70] H. ElSawy, E. Hossain, and M. Haenggi, “Stochastic geometry for modeling, analysis, and design of multi-tier and cognitive cellular wireless networks: A survey,” *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 3, pp. 996–1019, 2013.
- [71] Z. Gong and M. Haenggi, “Interference and outage in mobile random networks: Expectation, distribution, and correlation,” *Mobile Computing, IEEE Transactions on*, vol. 13, no. 2, pp. 337–349, 2014.
- [72] J.-F. Chamberland and V. V. Veeravalli, “How dense should a sensor network be for detection with correlated observations?,” *Information Theory, IEEE Transactions on*, vol. 52, no. 11, pp. 5099–5106, 2006.
- [73] E. Jondeau and M. Rockinger, “Gramcharlier densities,” *Journal of Economic Dynamics and Control*, vol. 25, no. 10, pp. 1457 – 1483, 2001.

REFERENCES

- [74] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Computing Survey*, vol. 41, no. 3, pp. 15:1–15:58, 2009.
- [75] L. Lamport, R. Shostak, and M. Pease, “The byzantine generals problem,” *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.
- [76] R. Santucci, M. K. Banavar, A. Spanias, and C. Tepedelenlioglu, *Estimation for amplify-and-forward transmissions with nonlinear amplifiers*. IET, 2013.
- [77] S. Dasarathan and C. Tepedelenlioglu, “Distributed estimation and detection with bounded transmissions over gaussian multiple access channels,” 2013.
- [78] T. E. Anderson, S. S. Owicki, J. B. Saxe, and C. P. Thacker, “High-speed switch scheduling for local-area networks,” *ACM Trans. Comput. Syst.*, vol. 11, pp. 319–352, Nov. 1993.
- [79] N. McKeown, “The islip scheduling algorithm for input-queued switches,” *IEEE/ACM Transactions on Networking*, vol. 7, pp. 188 –201, Apr. 1999.
- [80] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, “Integrated coverage and connectivity configuration in wireless sensor networks,” *SenSys’03: Proceedings of the First International Conference on Embedded Networked Sensor Systems*, pp. 28–39, 2003.
- [81] Y. Chen and Q. Zhao, “On the lifetime of wireless sensor networks,” *IEEE Communications Letters*, vol. 9, pp. 976 – 978, nov. 2005.
- [82] J. Deng, Y. Han, W. Heinzelman, and P. Varshney, “Scheduling sleeping nodes in high density cluster-based sensor networks,” *Mobile Networks and Applications*, vol. 10, no. 6, pp. 825–835, 2005.
- [83] I.-H. Li, I.-E. Liao, and F.-N. Wu, “An uncapacitated facility location based cluster hierarchy scheme on wireless sensor networks,” *WSEAS Transactions on Communications*, vol. 9, no. 11, pp. 740–757, 2010.

REFERENCES

- [84] S. Park, K. Shin, A. Abraham, and S. Han, "Optimized self organized sensor networks," *Sensors*, vol. 7, no. 5, pp. 730–742, 2007.
- [85] W. Zhou, H.-m. Chen, and W.-h. Xie, "Cluster merging algorithm with link optimization for wireless sensor networks," *International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, sept. 2006.
- [86] G. Gupta and M. Younis, "Load-balanced clustering of wireless sensor networks," *IEEE International Conference on Communications*, vol. 3, pp. 1848–1852 vol.3, may 2003.
- [87] M. Younis, K. Akkaya, and A. Kunjithapatham, "Optimization of task allocation in a cluster-based sensor network," *Eighth IEEE International Symposium on Computers and Communication*, pp. 329–334 vol.1, june-3 july 2003.
- [88] G. Gupta and M. Younis, "Fault-tolerant clustering of wireless sensor networks," *IEEE Wireless Communications and Networking*, vol. 3, pp. 1579–1584 vol.3, march 2003.
- [89] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, pp. 660–670, oct 2002.
- [90] V. Loscri, G. Morabito, and S. Marano, "A two-levels hierarchy for low-energy adaptive clustering hierarchy (tl-leach)," *IEEE 62nd Vehicular Technology Conference*, vol. 3, pp. 1809–1813, sept., 2005.
- [91] S. Lindsey and C. Raghavendra, "Pegasis: Power-efficient gathering in sensor information systems," *IEEE Aerospace Conference Proceedings*, vol. 3, pp. 3–1125–3–1130 vol.3, 2002.
- [92] L. Yu, N. Wang, W. Zhang, and C. Zheng, "Group: A grid-clustering routing protocol for wireless sensor networks," *International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–5, sept. 2006.
- [93] K. Shin, A. Abraham, and S. Han, "Self organizing sensor networks using intelligent clustering," *Computational Science and Its Applications - ICCSA 2006*, vol. 3983, pp. 40–49, 2006.

REFERENCES

- [94] H. Chan and A. Perrig, “Ace: An emergent algorithm for highly uniform cluster formation,” *Wireless Sensor Networks*, vol. 2920, pp. 154–171, 2004.
- [95] B. Deb, S. Bhatnagar, and B. Nath, “A topology discovery algorithm for sensor networks with applications to network,” *DCS Technical Report DCS-TR-441*, May 2001.
- [96] T. Voigt, A. Dunkels, J. Alonso, H. Ritter, and J. Schiller, “Solar-aware clustering in wireless sensor networks,” *Ninth International Symposium on Computers and Communications*, vol. 1, pp. 238 – 243, July 2004.
- [97] K. Kinoshita, T. Okazaki, H. Tode, and K. Murakami, “A data gathering scheme for environmental energy-based wireless sensor networks,” *5th IEEE Consumer Communications and Networking Conference*, pp. 719 –723, Jan. 2008.
- [98] C. Alippi, R. Camplani, C. Galperti, and M. Roveri, “A robust, adaptive, solar-powered wsn framework for aquatic environmental monitoring,” *IEEE Sensors Journal*, vol. 11, pp. 45 –55, Jan. 2011.
- [99] P. Zhang, G. Xiao, and H.-P. Tan, “Clustering algorithms for maximizing the lifetime of wireless sensor networks with energy-harvesting sensors,” *Computer Networks*, vol. 57, pp. 2689–2704, Oct. 2013.
- [100] H. Gou, G. Li, and Y. Yoo, “A partition-based centralized leach algorithm for wireless sensor networks using solar energy,” *Proceedings of the 2009 International Conference on Hybrid Information Technology*, pp. 60–66, 2009.
- [101] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, “Power management in energy harvesting sensor networks,” *ACM Trans. Embed. Comput. Syst.*, vol. 6, Sept. 2007.
- [102] C. Vigorito, D. Ganesan, and A. Barto, “Adaptive control of duty cycling in energy-harvesting wireless sensor networks,” *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 21 –30, june 2007.
- [103] D. Tian and N. D. Georganas, “A coverage-preserving node scheduling scheme for large wireless sensor networks,” *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pp. 32–41, 2002.

- [104] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Proceedings of the 7th annual international conference on Mobile computing and networking*, pp. 85–96, 2001.
- [105] J. Piorno, C. Bergonzini, D. Atienza, and T. Rosing, "Prediction and management in energy harvested wireless sensor nodes," *Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on*, pp. 6–10, may 2009.
- [106] D. R. Cox, "Prediction by exponentially weighted moving averages and related methods," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 23, no. 2, pp. pp. 414–422, 1961.
- [107] C. Moser, L. Thiele, D. Brunelli, and L. Benini, "Robust and low complexity rate control for solar powered sensors," *Design, Automation and Test in Europe, 2008. DATE '08*, pp. 230–235, march 2008.
- [108] N. Karunanithi, D. Whitley, and Y. Malaiya, "Using neural networks in reliability prediction," *Software, IEEE*, vol. 9, pp. 53–59, july 1992.
- [109] E. L. Lloyd and G. Xue, "Relay node placement in wireless sensor networks," *Computers, IEEE Transactions on*, vol. 56, no. 1, pp. 134–138, 2007.
- [110] X. Han, X. Cao, E. L. Lloyd, and C.-C. Shen, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 5, pp. 643–656, 2010.
- [111] R. Liu, I. J. Wassell, and K. Soga, "Relay node placement for wireless sensor networks deployed in tunnels," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on*, pp. 144–150, IEEE, 2010.
- [112] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang, "Two-tiered constrained relay node placement in wireless sensor networks: Computational complexity and efficient approximations," *Mobile Computing, IEEE Transactions on*, vol. 11, no. 8, pp. 1399–1411, 2012.

REFERENCES

- [113] S. Srinivasa and M. Haenggi, "Distance distributions in finite uniformly random networks: Theory and applications," *Vehicular Technology, IEEE Transactions on*, vol. 59, pp. 940–949, Feb 2010.
- [114] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A witness-based approach for data fusion assurance in wireless sensor networks," *IEEE Global Telecommunications Conference, 2003. GLOBECOM '03.*, vol. 3, pp. 1435–1439 vol.3, 2003.
- [115] D. Janakiram, V. Adi Mallikarjuna Reddy, and A. Phani Kumar, "Outlier detection in wireless sensor networks using bayesian belief networks," *First International Conference on Communication System Software and Middleware, 2006. Comsware 2006.*, pp. 1–6, 2006.
- [116] H. Van Trees, *Detection, estimation, and modulation theory.. part 1.. detection, estimation, and linear modulation theory*. Wiley New York, 1968.
- [117] M. B. De Kock, *Gaussian and non-Gaussian-based Gram-Charlier and Edgeworth expansions for correlations of identical particles in HBT interferometry*. PhD thesis, Stellenbosch: University of Stellenbosch, 2009.
- [118] T. S. Rappaport, *Wireless communications: principles and practice*. Prentice Hall PTR New Jersey, 2002.
- [119] D. W. Hearn and J. Vijay, "Efficient algorithms for the (weighted) minimum circle problem," *Operations Research*, vol. 30, no. 4, pp. pp. 777–795, 1982.
- [120] B. Calhoun, D. Daly, N. Verma, D. Finchelstein, D. Wentzloff, A. Wang, S.-H. Cho, and A. Chandrakasan, "Design considerations for ultra-low energy wireless microsensor nodes," *IEEE Transactions on Computers*, vol. 54, no. 6, pp. 727–740, 2005.
- [121] J. E. Hopcroft and R. M. Karp, "A $n^5/2$ algorithm for maximum matchings in bipartite," *12th Annual Symposium on Switching and Automata Theory.*, pp. 122–125, 1971.
- [122] K. Mjelde, "Max-min resource allocation," *BIT*, vol. 23, no. 4, pp. 529–537, 1983.
- [123] "Spatial point processes and their applications," *Stochastic Geometry*, vol. 1892, pp. 1–75, 2007.

REFERENCES

- [124] S. Kay, *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*. Prentice Hall PTR, 1998.
- [125] R. S. Targino, G. W. Peters, G. Sofronov, and P. V. Shevchenko, “Optimal insurance purchase strategies via optimal multiple stopping times,” *arXiv preprint arXiv:1312.0424*, 2013.
- [126] R. Askey and J. A. Wilson, *Some basic hypergeometric orthogonal polynomials that generalize Jacobi polynomials*, vol. 319. American Mathematical Soc., 1985.
- [127] M. G. Cruz, G. W. Peters, and P. V. Shevchenko, *Fundamental aspects of operational risk and insurance analytics*. Wiley, 2014.
- [128] D. P. Kroese and Z. I. Botev, “Spatial process generation,” *arXiv preprint arXiv:1308.0399*, 2013.
- [129] M. M. Fréchet, “Sur quelques points du calcul fonctionnel,” *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, vol. 22, no. 1, pp. 1–72, 1906.
- [130] D. Dowson and B. Landau, “The frechet distance between multivariate normal distributions,” *Journal of Multivariate Analysis*, vol. 12, no. 3, pp. 450–455, 1982.