Energy-efficient, and Reliable Communication in Wireless Sensor Networks

PhD DISSERTATION OF

Nguyen Thai Hoc, M.Sc.

Supervisor

Dr. Levendovszky János, D.Sc.
Doctor of the Hungarian Academy of Sciences

June 29, 2018
Acknowledgements

First and foremost, I would like to thank my supervisor Dr. Janos Levendovszky for giving me the opportunity to do research at Budapest University of Technology and Economics. His guidance, patience, and wisdom show me not only the way of doing research but also the way of being a decent and responsible society member. This thesis would not have finished without his help.

I would also like to thank Dr. Do Van Tien, who has not only been so kind as to discuss some of the problems presented in the thesis but also shared his vast knowledge and experience with me.

I would like to thank my committee members Dr. Peter Ekler and Dr. Kovacs Lorant, for their valuable feedbacks and comments on my research, from which I have benefited greatly.

I would also like to thank all other staff members of the Networked Systems and Services Department for their help and support during my study period.

I also want to thank the Stipendium Hungarian for funding my research.

I am very grateful to my colleges and my friends, who help me a lot in this research. Last but not least, I would like to express my greatest gratitude to my parents, my dear wife, Nguyen Thi Thu Hien and my children, Nguyen Thu Hong Ngoc and Nguyen Thu Lan Huong, for their unconditional love, support, understanding, and encouragement. I would not be able to achieve anything without their love and support. I am very aware of how lucky I am and want to thank you for always accompanying me on my way.
Executive Summary

Over the last few years, we have witnessed an increasing interest in Wireless Sensor Networks (WSNs). The combination of recent advances in robotics, digital electronics, wireless communications has enabled in many practical applications such as target tracking, military, health-care monitoring, environment monitoring, and so on [6]. Unfortunately, WSNs still face many challenges in data communication, mainly caused by limited energy capacity, limited storage capacity, short-range radio signal, dynamic routing protocol, and security problems.

To overcome these limitations, the foremost concerns of energy-efficient routing protocols in WSNs are to minimize transmission overhead, to minimize latency, to improve the system reliability and to prolong the network lifetime [95]. This dissertation has the same aims in two critically important research areas: (i) improving the energy efficiency and (ii) the reliability of communication in WSNs.

Many authors have tried to develop energy-efficient and reliable routing protocols for WSNs. However, there seems to be no research emphasizing the importance of the combination of energy efficiency and the reliability in designing a routing protocol. Therefore, I focus on improving both energy efficiency and system reliability in WSN. In my research, various data transmission techniques are used to develop an efficient communication in WSNs. My related results can be found in [51, 79, 81, 107].

I developed a Cost-Minimizing Scheduling (CMS) algorithm for data transmission in WSNs based on the Orthogonal Frequency Division Multiplexing (OFDM) system. In this way, my proposed algorithm achieves low system cost, low runtime complexity, while still guaranteeing a predefined probability of packet loss at the same time. In [107], I have combined the cluster-head election algorithm and the Mobile Sink (MS) trajectory optimization algorithm to propose the optimal MS movement strategy. Unlike typical algorithms in cluster head election, my algorithms tried to find the best location of a single CH for each cluster, where both requirements (i.e., minimum energy consumption and the energy balance among nodes in the network) would be satisfied. I also considered scenarios related to the MS trajectory in mobile wireless sensor networks. Herein, an optimal trajectory of the MS is obtained when both minimizing energy consumption and the constraint time in data gathering are met. In [79], I proposed a new algorithm, which finds the optimal paths from the Source Nodes (SN) to the Base Station (BS) in WSNs based on the Rayleigh fading model. Additionally, it is also proven that outlier detection technique is one of the most useful techniques for reducing the energy consumption as well as decreasing the transmission overhead in data communication. After analyzing carefully the advantages and disadvantages of some classical methods for detecting outliers, I developed a near-optimal method to detect outliers in the streaming sensor data with the short execution time and high accurate identification rate. For solving the localization problem in Non-Line-of-Sight (NLoS) environments, I exploited the Received Signal Strength Indicator (RSSI) to estimate the distance between the unknown position source nodes and the Moving Beacon (MB). I proposed the Simulated Annealing (SA) method wherein the next movement po-
sition of the MB is depended on the simulated annealing probability. In this way, the MB simply moves to next higher RSSI position. Unfortunately, if the MB reaches the plateau or local maximum points, the RSSI at the neighbor points will be equal to or even less than the signal at the current position of the MB. As a result, the MB may fail to reach the global maximum point. To tackle this problem, the MB may still move to the next position if the acceptance probability is higher than a random value in the range $[0, 1]$. As a conclusion, the results presented in my thesis contribute significantly to improving the energy-efficient and reliable communication in WSNs. They are widely used in a wide range of WSN applications.
Contents

Acknowledgements 1

Executive Summary i

List of Figures viii

List of Tables 1

1 Introduction 2

1.1 Problem Statement ........................................... 3

1.1.1 Resource management and packet scheduling ....... 3

1.1.2 Maximizing the network lifetime in Mobile Wireless Sensor Network 4

1.1.3 Improving QoS routing for WSNs .................. 4

1.1.4 Outlier detection in WSNs data .................. 5

1.1.5 Position location technique in Non-Line-of-Sight environment for WSNs ........................................... 5

1.2 The objectives of the dissertation .................... 6

1.3 Main Contributions .......................................... 6

1.3.1 Resource management and packet scheduling ....... 6

1.3.2 Maximizing the network lifetime in Mobile Wireless Sensor Network 7

1.3.3 Improving QoS routing for WSNs .................. 7

1.3.4 Outlier detection in WSNs data .................. 7

1.3.5 Position location technique in Non-Line-of-Sight environment for WSNs ........................................... 7

1.4 Structure of the dissertation ........................ 7

2 A new scheduling algorithm for energy-aware and reliable data transmission in WSNs 9

2.1 Introduction ................................................ 9

2.2 Problem formulation ....................................... 10

2.3 Solution by random scheduling ....................... 15
2.4 The proposed algorithm ........................................... 16
2.5 Numerical results .................................................. 19
   2.5.1 Experimental scenarios ...................................... 19
   2.5.2 Comparative performance analysis ......................... 20
2.6 Conclusions and directions for future research ................. 27

3 An Efficient Approach for Maximizing Lifespan in Wireless Sensor Networks by Using Mobile Sinks 28
   3.1 Introduction ...................................................... 28
   3.2 System Model and Assumptions ................................ 30
      3.2.1 Basic Assumptions ......................................... 30
      3.2.2 Network Model ............................................. 31
      3.2.3 Energy Model .............................................. 31
   3.3 Proposed Approach ............................................. 33
      3.3.1 Cluster Heads Election Algorithm ......................... 33
      3.3.2 Optimizing the Trajectory of the Mobile Sink .......... 35
      3.3.3 Optimal Movement Strategy for Mobile Sink .......... 40
   3.4 Performance Evaluation and Discussion ....................... 40
      3.4.1 Simulation Environment .................................... 41
      3.4.2 Numerical Results and Discussion ....................... 41
   3.5 Conclusions ..................................................... 44

4 Quality-of-Service Routing Protocol for wireless sensor networks 46
   4.1 Introduction ...................................................... 46
   4.2 The model ........................................................ 47
   4.3 A novel reliable routing algorithm to maximize the lifespan 49
      4.3.1 Characterization of the energy state of the network 49
      4.3.2 Novel routing algorithm .................................... 49
   4.4 Complexity analysis ............................................. 51
   4.5 Numerical results ................................................. 51
      4.5.1 Performance analysis and numerical results for HQRA algorithm 51
   4.6 Conclusions and directions for future research ............ 57

5 Prediction-based outlier detection for wireless sensor networks 58
   5.1 Introduction ...................................................... 58
   5.2 Related works .................................................. 59
   5.3 Outlier detection methods ...................................... 60
      5.3.1 Hampel Identifier .......................................... 60
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.2 The decision theoretic framework to detect outliers</td>
<td>62</td>
</tr>
<tr>
<td>5.4 Application of the outlier detection</td>
<td>64</td>
</tr>
<tr>
<td>5.4.1 Numerical analysis on the observed data given in [28]</td>
<td>64</td>
</tr>
<tr>
<td>5.4.2 Numerical analysis on synthetic data</td>
<td>64</td>
</tr>
<tr>
<td>5.4.3 Numerical analysis on the real dataset</td>
<td>68</td>
</tr>
<tr>
<td>5.4.4 Numerical analysis on detecting the network violations in the 1999 DARPA dataset</td>
<td>69</td>
</tr>
<tr>
<td>5.4.5 Numerical analysis on detecting outliers in the monthly series of the Italian Industrial Production Index from 1981 to 1996</td>
<td>71</td>
</tr>
<tr>
<td>5.4.6 Numerical analysis on detecting online attacks</td>
<td>71</td>
</tr>
<tr>
<td>5.5 Conclusions and directions for future research</td>
<td>76</td>
</tr>
</tbody>
</table>

| 6 Position Location technique in Non-Line-of-Sight Environments for Wireless Sensor Networks | 77   |
| 6.1 Introduction                                                       | 77   |
| 6.2 Related works                                                      | 78   |
| 6.3 The model                                                         | 79   |
| 6.3.1 System Environments and Assumptions                             | 79   |
| 6.3.2 RSS-Based distance estimation                                   | 80   |
| 6.4 Localization algorithms                                            | 81   |
| 6.4.1 Localization by Expected Position Circle Scan (EPCS)            | 81   |
| 6.4.2 Solution by steepest ascent search (SAS)                         | 83   |
| 6.4.3 Stochastic gradient ascent (SGA) method for localization in WSNs | 85   |
| 6.4.4 Local obstacle avoidance for the MB in the NLoS environments     | 86   |
| 6.4.5 Simulated annealing method for approximating the global maximum | 89   |
| 6.5 Performance evaluation                                            | 89   |
| 6.5.1 Simulation design                                               | 89   |
| 6.5.2 Simulation results                                              | 92   |
| 6.6 Conclusions and directions for future research                    | 94   |

| 7 SUMMARY OF RESULTS AND THESES                                        | 96   |
| 7.1 Summary                                                           | 96   |
| 7.2 Future research plan                                              | 98   |

Abbreviations 101

Bibliography 113

Appendix 114
A.1 Proof of Lemma 1 in section 6.4 [35] . . . . . . . . . . . . . . . . . . . . . . . . . 114
A.2 Proof of $R_s(t) \leq R_{\text{max}}$ . . . . . . . . . . . . . . . . . . . . . . . . . 114
List of Figures

2.1 Block diagram of a multiuser OFDM transmitter with adaptive sub-carrier and power allocation [132]. ........................................ 11
2.2 Sub-carrier and Power allocation in WSN. ............................... 13
2.3 Illustration of data transmission between node $k$th and the BS ........ 14
2.4 The flow diagram of the RS algorithm ...................................... 15
2.5 The comparison of system cost .............................................. 21
2.6 The elapsed time of algorithms ............................................ 21
2.7 The improvement in system cost ............................................ 23
2.8 Improving the reliability of data transmission with $(X, M) = (25, 10)$ . 25
2.9 The comparison of data rate in data communication ..................... 26
2.10 Fairness pointer vs number of nodes .................................... 27

3.1 Structure of communication of the WSNs ................................. 32
3.2 Format of the “HELLO” message broadcast by the sensor nodes ........ 33
3.3 Cluster communication mechanism if node $S_1$ becomes $A_1$’s CH node . 35
3.4 Sink movement strategy for Scenario 1. ................................... 38
3.5 Sink movement strategy in Scenario 2 by OMS algorithm ............... 40
3.6 Network lifetime comparison ............................................... 44
3.7 Network lifetime comparison with different network sizes ............... 44
3.8 Network lifetime variation with different node densities ................. 45

4.1 Multi-hop communication between sensor nodes and the BS in WSNs . . 48
4.2 Choose the smallest transmission energy of the path nodes by HQRA . . 51
4.3 The optimizing process of the HQRA algorithm .......................... 52
4.4 The optimal path by PEDAP algorithm .................................... 53
4.5 The optimal path by HQRA algorithm .................................... 54
4.6 The comparison of the energy balancing among nodes in the network . . 55
4.7 Comparisons of the network lifetime between PEDAP algorithm and HQRA algorithm ........................................ 55
4.8 The Network reliability with different network sizes .......................... 56
4.9 The trend of network lifetime when increasing the reliability threshold .......................... 56

5.1 Possible outcomes of a certain outlier detection algorithm .................. 61
5.2 Time-series data with $k = 1000; L = 3$, and addition outliers ............... 67
5.3 The number of identified outliers by the HI algorithm ......................... 67
5.4 The identified outliers by the ODPOE algorithm ................................ 68
5.5 The identification rate in a time-series data ................................. 68
5.6 Outlier detection in temperature values on March 6th, 2004 .................. 69
5.7 Outlier detection in humidity values on March 6th, 2004 ...................... 69
5.8 Outlier detection in light values on March 6th, 2004 ........................... 70
5.9 Outlier detection in the Italian industrial production index 1981–1996 .... 71
5.10 The experimental setup of the network ....................................... 74
5.11 Detection of data-flow anomaly ............................................... 75
5.12 Outlier detection in traffic flow within the network .......................... 75
5.13 The Bandwidth used by Users in the network ................................. 76

6.1 The correlation of received signal strength and separation distance ........ 82
6.2 Position estimation with moving beacon in WSN ............................... 84
6.3 Position detection by EPCS algorithm ........................................ 84
6.4 Position detection by SAS algorithm ........................................... 86
6.5 The disadvantages of the steepest ascent method ............................... 88
6.6 The trajectory of the MB by SGA algorithm and SA algorithm ............. 90
6.7 The flowchart of the SA algorithm .............................................. 91
6.8 The location error of each sensor node ...................................... 93
6.9 The ALE and AET versus the velocity of the MB ............................. 95

A.2.1 The reduced length of the MS’s trajectory after changing the CHs’ transmission ranges .......................... 115
List of Tables

1.1 Summary of research areas ............................................. 6
2.1 The specification of a packet assignment scheme ..................... 17
2.2 Moving a packet from sub-carrier $S_H$ to sub-carrier $S_{H+1}$ .......... 17
2.3 Sending a new packet in sub-carrier $S_1$ .......................... 18
2.4 The improvement in system cost with $M = 10$ ..................... 22
2.5 The improvement in system cost with $M = 15$ ..................... 22
2.6 The improvement in system cost with $M = 20$ ..................... 24
2.7 The improvement in system cost with $M = 50$ ..................... 24
2.8 The improvement in system cost with $M = 100$ ................... 24
3.1 Example for cluster-head election .................................... 34
3.2 The settings of simulation parameters ............................... 43
3.3 Network lifetime (rounds) by CHE algorithm with different values of $(\alpha, \beta)$ 43
4.1 The setting of simulation parameters .................................. 53
4.2 The Probability of successful packets with the smallest transmission energy 54
5.1 Possible outcomes of outlier detection method ..................... 60
5.2 The observed data in [28] ............................................. 64
5.3 Outlier detection by the HI algorithm ............................... 65
5.4 The process of outlier detection by the ODPOE algorithm ............. 66
5.5 Comparison of the identification rate in outlier detection ............ 66
5.6 Outliers in dataset from Intel Berkeley Research lab on March 6th, 2004 70
5.7 The network violations in DARPA dataset ........................... 72
5.8 The network violations in fourth and fifth week of the 1999 DARPA dataset 73
5.9 Outlier detection in the Italian industrial production index 1981–1996 73
6.1 The settings of simulation parameter .................................. 91
6.2 The performance of localization algorithms based on the ALE and the AET 94
7.1 Summary of my theses .................................................. 97
7.2 Heuristic solutions to overcome the limitations of mobility models  .... 99
Present day info-communication technologies are in great need for algorithms, which can optimize and further extend the network performance. There are several platforms and application domains (e.g., Internet of Things (IoT), Artificial Intelligence (AI) Cloud Computing) whose performance requires novel optimization methods in improving the reliability, outlier detection, and Quality-of-service (QoS) routing in data transmission. To be known as a subset of IoT, the WSN consists of hundreds or even thousands of small, inexpensive wireless nodes that are deployed in sensing field. These wireless nodes are able to sense events and communicate with neighbor nodes by wireless connectivity with high economic value. Furthermore, WSN may be deployed in some rough terrains where some traditional IoT devices cannot work properly. For example, some endpoint devices in IoT (e.g., security cameras, mobiles, smart watches, computers, cars, robot, industrial machines) cannot connect to a network without supporting of well-designed infrastructure. However, by using some special routing protocols, each node in WSN is capable of sensing its environment, locally processing data, and transmitting data to the collection devices with high reliability and in short execution time. As a result, WSNs have been emerged as one of the most promising technologies for a number of applications, such as military applications, industrial applications, agricultural applications, mobile health applications, as well as numerous consumer applications. By using a WSN, we can sense and collect data from the world around us for environmental monitoring, healthcare monitoring, controlling, managing systems or many other purposes. However, despite many years of development, the WSN still faces many challenges, mainly caused by its constrained resources such as limited power, short radio transmission range, narrow radio bandwidth, and limited memory capacity. These limitations directly affect the network performance including network lifetime, network reliability, and Quality of Service (QoS) in real-world applications. Therefore, the critical requirements of these applications are low data transmission cost, long network lifespan, and high reliability communication. To satisfy these requirements, a large number of works has been done in a WSN, such as to minimize the energy consumption, to maximize the network lifetime, to increase the probability of successful packets in data transmission, as well as, to detect outliers in raw sensing data from various wireless sensor nodes.

The objective of my research is to provide new and efficient solutions, which minimize the system cost, decrease the transmission overhead, improve the network lifespan and maximize the system reliability in WSNs. These solutions are described in solving the problems in five specific domains:
(i) the problem of finding the optimal resource management and packet scheduling in WSNs;

(ii) the problem of maximizing the network lifetime in Mobile Wireless Sensor Network (MWSNs);

(iii) the problem of improving the QoS routing;

(iv) the problem of prediction-based outlier detection for WSNs;

(v) and localization problem in Non-Light-of-Sight environments.

Although these five domains may seemingly be classified into different areas, they altogether have a great impact on the efficiency of WSNs. The algorithms and methods proposed in my dissertation can also be used in other applications such as public transport control and guidance systems, financial computational systems, etc.

In the remainder of this chapter, a brief summary of the technological background, motivations and state of the art of each topic are discussed in Section 1.1. Section 1.2 describes the objectives of this work. Afterwards, in Section 1.3 I summary the main contributions of this thesis, before turning to the structure of my dissertation in Section 1.4.

1.1 Problem Statement

In this section, I provide an overview of the motivation and real-world applications of the selected problems.

1.1.1 Resource management and packet scheduling

It is known that wireless sensor nodes are powered with small batteries with low battery power. These batteries would be difficult or even impossible to recharge or replace in some special environments. It is also proven that a sensor node will spend more than 50% of its energy for communication activity [158]. Therefore, minimizing energy consumption in data communication is one of the primary concerns in WSNs applications. Additionally, “Timely delivery” and “Guaranteed delivery” [103, 113] are also important communication properties in WSNs. To satisfy these requirements, in data communication, sensing data should be collected at the BS with minimum system cost, while guaranteeing a given probability of successful data transmission. To tackle this problem, utilizing OFDM system is one of the potential solutions. By OFDM system, a high-speed data is divided into several slower rate signals, and then transmits each slower rate signal in separate frequency bands [103]. In OFDM system, the total transmission time is divided into radio frames and each radio frame contains time-slots with hundreds of sub-carriers in one time-slot. The objective is to make an effective resource allocation scheme, by which all sensor nodes transmit their data packets to the BS with short timely delivery and with the lowest cost data transmission. The sub-channel assignment and power allocation based on OFDM are widely used in the real-world applications. It may be used for down-link radio transmission by 3GPP and many other applications such as digital subscriber loops, wide area broadcasting and local area networks [103]. In [33], the deployments of the OFDM technology are further in the cellular mobile radio standard 3GPP Long Term Evolution (LTE) and future broad wireless access standard such as IEEE 802.16x, WiMAX. In [124], the discontinuous OFDM is used for dynamic spectrum access in idle TV channels. Unfortunately, the number of spectrum fragments in that study are limited [62] and tied together for
complicating the design of protocols [52]. The optimal paths by using transmission power adaption-based routing technique have been described in numerous works [36, 57, 70, 83]. However, most of these research concentrate on improving the performance in sub-carrier and power allocation schemes without paying enough attention to reliability issues.

1.1.2 Maximizing the network lifetime in Mobile Wireless Sensor Network

Recently, there has been the rapid development of robotics, sensor structure, and wireless communication techniques make feasible to improve the network lifetime in Mobile Wireless Sensor Networks (MWSNs). Moreover, the MWSNs have provided many benefits such as minimizing system cost, enhancing the connectivity, increasing the coverage, improving the reliability, as well as achieving high energy efficient data transmission. Recent research has shown that MWSNs have a wide range of applications [104], especially in healthcare applications [24, 37, 73, 74], industrial applications [31, 114, 115], agriculture applications [9, 41, 122], transportation applications [89, 137, 153], home applications [38, 155], and in military applications [93, 152]. Many authors [5, 39, 86, 105, 148] have tried to extend the network lifetime by mobile devices in WSN over the years. As proposed in [86], the network lifetime is improved by a non-uniform deployment achieved by a moving algorithm. In their techniques, mobile sensors move to appropriate locations for maintaining the network coverage and prolonging the network lifetime. However, in this way, the energy consumed for moving is significantly big, and the movement of mobile sinks is not feasible in Non-Light-of-Sight environments. The other technique in MWSNs to improve the network lifetime is given in [105]. In that study, a mobile sink based protocol (MSRP) for WSNs is proposed. The main idea of the MSRP is described as follows. In every cycle, the sensing data from member cluster nodes will be collected by a Cluster-Head node (CH). The Mobile Sink (MS) will move to the locations of Cluster-Head nodes (CHs) to aggregate data. During gathering data time, the MS also collects the residual energy information of CHs for its movement in next phase. After collecting all residual energy information of all CHs in the network, the MS will move to the location of CH which has the highest residual energy in the next phase. In this way, the MSRP avoids completely the hotspot problem in WSN and it also can improve the network lifetime in the network. Unfortunately, the sensed data will be lost by buffer overflow at the CHs, which have lower residual energy and are far from the current location of the MS. Although all of above techniques have achieved their contributions in improving the network lifetime, none have really succeeded in proposing a successful operating cycle of a MWSN wherein a signal is generated, collected, and analyzed.

1.1.3 Improving QoS routing for WSNs

In order to enable users to monitor accurately any position in the sensing field, hundreds or thousands of microsensor nodes need to transmit their sensing data or forward the data from their neighbor nodes toward the BS within its radio transmission range. These networks require efficient routing protocols, which are energy efficient [50, 116], having low latency [10, 111], having high reliability [34, 94], very secure [21, 133], and they are able to alleviate and handle the bottleneck problems in WSNs [55, 126]. Additionally, routing protocol plays crucial role in many fields especially in improving Quality-of-Service in WSNs. As a consequence, a great number of authors (i.e., Low-energy adaptive clustering hierarchy (LEACH) [50], Power-Efficient Gathering in Sensor Information Systems (PEGA-SIS) [127], and (Power Efficient Data gathering and Aggregation Protocol (PEDAP) [144])
have made efforts in order to enhance the network lifetime based on effective and energy-aware protocols. However, none of them takes into account of providing energy balancing under reliability constraints. Furthermore, there seems to be no clear technique emphasizing the importance of combination between energy efficiency (minimized energy consumption and energy balance among nodes in the network) and reliability constraints (maintaining the system reliability in the network) in a routing protocol. They are both critical features in WSN routing protocols [134], therefore, an efficient routing protocol for WSNs should satisfy these features.

1.1.4 Outlier detection in WSNs data

In some applications, huge of data from hundreds or even thousands of tiny nodes will be received by the BS. However, due to constraints on signal processing and communication capabilities of WSNs, there are some unusual data (called outliers) may result from sensor malfunction, process disturbances, human-related errors, and/or a sudden change in the state of the environment. It is well-known that outliers detection is one of the most important preprocessing steps in any data analytical applications. The outliers might seriously affect the accuracy of data analysis which causes the model misspecification. Therefore, many authors have tried to detect the outliers in order to improve the quality of collected data in WSNs. In the literature, the Hampel Identifier (HI) algorithm is the most widely used which provides an efficient outliers identifier [44]. However, the HI reveals its limitations when working with highly auto-correlated data process. More precisely, it may fail to capture outliers due to the strong autocorrelation [87]. Additionally, in HI algorithm, the standard deviation estimates are replaced by the MAD from the median. Unfortunately, this MAD scale estimator can behave badly with coarsely quantized data [119]. Despite its importance, most of existing outlier detection methods are still mainly designed for cleaning data and do not take into account the real-time outliers identification problem. This may seriously affect the accuracy of real-time decision making. To avoid the risks of anomalies data, an outlier detection method should identify outliers in the streaming data with the high accuracy rate. The outcomes of an outlier detection method can help identify the abnormal and irregular patterns hidden in huge datasets, which reduce the energy consumption, as well as minimize the memory usage.

1.1.5 Position location technique in Non-Line-of-Sight environment for WSNs

Localization of every sensor node in the network plays a critical role in many WSN applications such as in (i) coverage, (ii) event detection, and routing designing [56,100,143,159]. This field has stimulated many researchers which resulted in various proposals on improving the accuracy of the location estimation. However, depending on the way of obtaining the distance information, they can be classified into two main types of position location techniques: (i) range-free type, and (ii) range-based type. The range-free type [18,109] is known as an economical technique because it only uses some reference points, counting the number of hops communication between unknown position node to the anchor nodes, or using some special protocol to locate sensor nodes. Without measuring distances or angles between node to node, this technique can save the energy consumption and no need to equip some expensive devices for the position detection. However, the range-free is not a reliable type because the error range between estimated and real positions may be felt in the interval 20% - 40% [140]. In contrast to the Range-free type, by the range-based technique, the locations of sensor nodes are estimated by using distances or
angles node-to-node methods (e.g., Time of Arrival (TOA) [22], Time Difference of Arrival (TDOA) [22, 156], Angle of Arrival (AOA) [108], Direction of Arrival (DOA) [101], and Received Signal Strength Indicator [143]). Although the range-based techniques give higher accuracy than the range-free techniques, they are still expensive techniques and need some more specific hardware for location detection. Moreover, these techniques may fail in the presence of obstacles in NLoS environments.

1.2 The objectives of the dissertation

Concluding the previous section, the objective of the dissertation is to develop algorithmic and tools, which are given in Table 1.1

<table>
<thead>
<tr>
<th>Research area</th>
<th>WSN applications</th>
<th>Further application of results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource management and packet scheduling</td>
<td>Packet scheduling, Lan protocols.</td>
<td>Telecommunication, Schedule of computational resources such as the field of financial analysis, or regulator of public transport.</td>
</tr>
<tr>
<td>Maximizing the network lifetime in Mobile Wireless Sensor Network</td>
<td>Cluster head election in WSN, Efficient mobility technology in WSNs</td>
<td>Optimize movement schedule.</td>
</tr>
<tr>
<td>Improving QoS routing for WSNs</td>
<td>Efficient routing technique for WSN</td>
<td>Quality of Service in telecommunication networks.</td>
</tr>
<tr>
<td>Outlier detection in WSNs data</td>
<td>Detecting outlier values and events in sensor readings.</td>
<td>Detecting network violations, outlier detection in time series data for some realistic monitor or tracking applications.</td>
</tr>
<tr>
<td>Position location technique in NLoS environments for WSNs</td>
<td>Tracking sensor location, find the best routing for WSNs based on nodes’ location</td>
<td>Traffic tracking applications, robotic strategies, events detection.</td>
</tr>
</tbody>
</table>

1.3 Main Contributions

In this section, a summary of main contributions for each selected topic to developing an energy efficient and reliable communication in WSNs is given as follows.

1.3.1 Resource management and packet scheduling

In the first thesis, I investigated the optimal resource management and packet scheduling in WSN communication. I introduced a new efficient method to transmit data with low energy consumption by developing a smart scheduler. The results demonstrate that my proposal outperforms other typical methods in terms of system reliability as well as the execution time. My algorithm can work well in WSNs by using OFDM systems, which is a potential candidate for 4G and some future mobile communications standards. My
method may be widely used in some applications of WSN as well as in financial analysis, or public transport allocation.

1.3.2 Maximizing the network lifetime in Mobile Wireless Sensor Network

In the second thesis, I investigated how data is collected and how to plan the trajectory of the MS in order to gather data in time with small energy consumption and long network lifetime. I proposed a new algorithm to find the optimal trajectory of mobile Sinks, by which the energy consumption and running time in a closed operating cycle of a WSN are minimized. The results of this topic can be used for robotic applications and some applications of traveling salesman problem.

1.3.3 Improving QoS routing for WSNs

In the third thesis, I investigate the QoS routing protocols for WSNs. I proposed a new algorithm to find the optimal path routing from the source node to the BS, by which a packet data is received successfully at the BS with minimum energy consumption under a reliability constraint. This approach can be used for designing an energy efficient communication protocol.

1.3.4 Outlier detection in WSNs data

In the fourth thesis, I investigated the outlier detection in the streaming data. I developed and proposed a new outlier detection method, which is based on the probability of the First Order Error (FOE). My proposed algorithm can detect and remove outliers on-line from measurement records of wireless sensors. Moreover, this method can be used for detecting the network violations or in some tracking applications.

1.3.5 Position location technique in Non-Line-of-Sight environment for WSNs

In the fifth thesis, I focused on developing some position location techniques in NLoS environments, which not only achieve high accuracy in location detection but also are reliable techniques with small execution time. In real-world applications, it may be used in traffic tracking and events detection.

1.4 Structure of the dissertation

The remainder of this dissertation is organized as follows: In Chapter 2, I define and study the problem of finding the optimal resource management and packet scheduling for WSNs. After analyzing the requirement of high reliability and high data-rate communications in WSNs, I present several similar existing techniques with their contributions as well as the challenges need to be solved. Based on the OFDM system, I then propose a Cost-Minimizing Scheduling (CMS) for data transmission in WSNs. Finally, I present the results of my method and the performance comparison between my algorithm and some typical algorithms. Chapter 3 defines and studies the problem of maximizing the network lifetime in Mobile Wireless Sensor Network. I propose a new cluster head election algorithm by
which one candidate node becomes a cluster head for a current round based on the balance between energy residual and energy consumption. I also propose optimal trajectory of the MS to collect all sensing data in the network within the reporting time. The simulation results demonstrate that my proposed algorithms achieve better performance than some other well-known algorithms. The problem of improving the QoS routing in WSNs is defined and studied in Chapter 4. In that chapter, firstly, I provide a brief review of the existing algorithms to improve QoS routing in WSNs. Then, I propose a new routing algorithm, which is able to find near-optimal paths with the smallest energy consumption and guarantee a given reliability. Finally, I present the simulation results of my algorithm as well as the comparison between my algorithm and some typical algorithms. Chapter 5 defines and studies the problem of outlier detection in streaming sensing data. I propose a new outlier detection method which is based on the probability of the FOE to detect and remove outliers on-line from sensing datasets. The performance of my algorithm will be evaluated in some real-world applications. In Chapter 6, I define and study the localization problem in WSNs. By utilizing a Moving Beacon to detect the positions of wireless sensor nodes, I propose some efficient approaches for sensor node localization. These approaches are also tested in some real WSN environments. The main results of the dissertation are summarized in Chapter 7. I also draw some general conclusions regarding the methodology used, and present directions for future research. Finally, in the appendices, I formally define my notation and abbreviations as well as presenting a list of my own publications and a full bibliography for this work.
Chapter 2

A new scheduling algorithm for energy-aware and reliable data transmission in WSNs

In this chapter, a novel algorithm is proposed to provide a Cost-Minimizing Scheduling (CMS) for data transmission in wireless sensor networks. The multi-carrier scheduling algorithm dynamically finds the optimal schedule for data transmission with minimum system cost, while ensures a given probability of successful data transmission to the Base Station. This approach does not only reduce the energy consumption of each sensor node by allocating effectively the amount of data to the corresponding sub-carriers but also achieves a given reliability with a minimum redundancy of transmitted data. The performance of the algorithm is analyzed over a wide range of parameters such as the number of sent packets, the number of sub-carriers needed to transfer data from each sensor node, and the required reliability of data transmission. The numerical results show that the proposed scheduling algorithm achieves low system cost and a low runtime complexity while guaranteeing a predefined probability of packet loss at the same time. The results also demonstrate that my CMS algorithm can work well in WSNs by using OFDM systems.

2.1 Introduction

One of the primary concerns of present day Wireless Sensor Networks technology is the safe data transmission and real-time data transmission. They are known as “Timely delivery” and “Guaranteed delivery” [33,103,113] of the data transmission. These properties are really important for remote monitoring systems and control systems which require high reliability and high data-rate communications. However, high data-rate communications are significantly limited by InterSymbol Interference (ISI). Hence, the applications of multi-carrier systems have rapidly increased in recent years as a potential solution to these problems. Orthogonal Frequency Division Multiplexing system is known as one special kind of multi-carrier transmission technology [33], which provides high data-rate transmission and widely used in wireless applications. The basic idea of the OFDM technique is to divide a high-speed data into several slower rate signals and then transmit each slower rate signal in separate frequency bands [103]. In OFDM system, the total transmission time is divided into radio frames and each radio frame contains time-slots.
with hundreds of sub-carriers in one time-slot. The objective is to make an effective
resource allocation scheme, which transmits successfully the packets from all nodes to
the BS, subject to shortening the delivery time and minimizing the system cost of data
transmission. As proposed in [139], there are two kinds of resource allocation scheme:
(i) fixed resource allocation; and (ii) dynamic resource allocation. The fixed resource
allocation schemes (e.g., Time Division Multiple Access (TDMA) and Frequency Division
Multiple Access (FDMA)) are the schemes, which fix the number of time-slots or the
number of sub-carriers when assigning them to the nodes [77]. In contrast to the fixed
resource allocation schemes, the dynamic resource allocation schemes [61] are known as
more flexible and more effective schemes when they allocate the resources adaptively (the
number of time-slots or sub-carriers) to each node dependent on its channel gains. In
this chapter, I propose a dynamic resource allocation scheme by assigning sensed data to
sub-carriers, which can be used in OFDM system.

Several authors have already dealt with the problem of transmission power adaption for
the multi-user OFDM system in an up-link transmission. In [70], the authors attempted
to maximize the rate-sum capacity by joining sub-carrier and power allocation in the
up-link of an Orthogonal Frequency-Division Multiple Access (OFDMA) system. In [83],
the sum-rate optimality of OFDMA in up-link multi-carrier systems has been studied.
The authors in [83] have shown that the number of shared sub-channels under the optimal
solution must be less than the number of total nodes. In [57], Huang, Jianwei, et al. used
a gradient-based scheduler for resource allocation scheme in the up-link OFDMA network.
They successfully allocated the physical layer resources (bandwidth and power) in order
to provide long-term QoS, guaranteed by the time-varying gradient of a utility function.
Finally, in [36], a low-complexity sub-carrier, power and rate allocation algorithm for the
OFDMA up-link was proposed. In that work, the authors have focused on the fairness
among users in order to maximize the sum rate under individual rate and transmit power
constraints.

However, most of the above algorithms failed to provide guarantees for reliable commu-
nication, where the aim is to receive a given number of data packets at the BS with a
predefined probability [80]. Thus, in this chapter, I propose a polynomial complexity
scheduling algorithm which guarantees reliable information transmission to the BS in
terms of minimizing the system cost when transmitting a given amount of data. By com-
paring my simulation results to those obtained by several previous algorithms, it turns
out that my proposed algorithm works more efficiently, achieves a predefined reliability
under a smaller system cost.

2.2 Problem formulation

In this section, I give a formal definition of multiuser adaptive OFDM system in WSNs.
The configuration of our multiuser adaptive OFDM system is shown in Figure 2.1. Let
us assume that the system has \( K \) nodes and a single Base Station. According to the
data transmission rule, data packets from different nodes will be allocated to different
sub-carriers. Therefore, the serial data from \( K \) nodes are allocated to \( M \) sub-carriers in
\( L \) time-slots. The focus of this work is to schedule the number of packets to be assigned
to a sub-carrier with minimum transmission power and subject to reliability constraints.
Figure 2.2 describes the sub-carrier and power allocation scheme in OFDM system. Herein,
\( K \) nodes share \( M \) sub-carriers in \( L = 20 \) time-slots of one radio frame. More precisely, I
summarize the features of the multiuser adaptive OFDM system as follows:
Each node from the set $U = \{u_i, i = 1, ..., K\}$ has a data rate $R_i$ packet per an OFDM symbol, and each node can use different sub-carriers. But no sub-carrier will be allowed to be shared by different nodes. Hence, I define $h_{mk} = 1$ if sub-carrier $m^{th}$ is allocated to node $k^{th}$, otherwise $h_{mt} = 0, \forall t \neq k$; where $\{t = 1, ..., K; m = 1, ..., M\}$.

The number of packets $y_{mk}$ are assigned to the sub-carrier $m^{th}$ by node $k^{th}$ is in the range $[0, ..., V]$ where $V$ is the maximum number of information packets/OFDM symbol that can be transmitted by each sub-carrier. In case $y_{mk} \neq 0$ then $y_{tk} = 0, \forall t \neq m$.

In order to maintain the required QoS at the receiver, the transmission power for allocating the packets to the sub-carrier $m^{th}$ is $c_m$. So the total transmission power of the system (the system cost) is computed as follows:

$$C_T = \sum_{k=1}^{K} \sum_{m=1}^{M} c_m y_{mk}. \quad (2.1)$$
• The data rate of the node $k^{th}$ is calculated as $[40, 132]$

$$R_k = \frac{B}{M} \sum_{m=1}^{M} h_{mk} \log_2(1 + \gamma_{mk}) \quad (2.2)$$

where $B$ denotes the total bandwidth system and $\gamma_{mk}$ is the signal-to-noise (SNR) of the sub-carrier $m^{th}$ for the node $k^{th}$ and is written as

$$\gamma_{mk} = \frac{c_{ml_{mk}}^2}{N_0 B M} \quad (2.3)$$

where $l_{mk}$ is the channel gain, and $N_0$ is the power spectral density of Additive White Gaussian Noise (AWGN).

• Thus the data rate in (2.2) can be transformed into

$$R_k = \frac{B}{M} \sum_{m=1}^{M} h_{mk} \log_2\left(1 + \frac{c_{ml_{mk}}^2}{N_0 B M}\right). \quad (2.4)$$

The general form of the sub-carrier and power allocation problem in a multi-user OFDM system is then given below.

Objective:

$$R_T = \max_{h_{mk}, c_m} \left\{ \frac{B}{M} \sum_{k=1}^{K} \sum_{m=1}^{M} h_{mk} \log_2 \left(1 + \frac{c_{ml_{mk}}^2}{N_0 B M}\right) \right\} \quad (2.5)$$

or

$$C_T = \min_{h_{mk}, c_m} \left\{ \sum_{k=1}^{K} \sum_{m=1}^{M} c_{ml_{mk}} \right\} \quad (2.6)$$

subject to:

$$C1 : h_{mk} \in \{0, 1\}, \forall k, m$$

$$C2 : \sum_{k=1}^{K} h_{mk} = 1, \forall k$$

$$C3 : \sum_{k=1}^{K} \sum_{m=1}^{M} h_{mk} = M, \forall k, m \quad (2.7)$$

$$C4 : c_{m} \geq 0, \forall m$$

$$C5 : \sum_{k=1}^{K} \sum_{m=1}^{M} c_{ml_{mk}} \leq P_{total}$$

where $P_{total}$ denotes the power constraint on the total transmission power of the system $C_T$.

In this chapter, based on the sub-carrier and power allocation problem in a multi-user OFDM system, my main objective is to minimize the total transmission power by developing a smart packet scheduling algorithm. Without the loss of generality, let us assume that data transmission from sensor nodes to the BS can use a channel with $M_f$ different carrier frequencies. Upon a query from the BS, $M$ sub-carriers are allocated to a node to transmit its packets. Additionally, different sub-carriers provide different qualities of transmission (the first sub-carrier is interference free as other nodes withheld these trans-
One bit data from sensor node $k$th

<table>
<thead>
<tr>
<th>Sensor node ID</th>
<th>Sub-carrier ID</th>
<th>Probability</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node 1</td>
<td>$S_1$</td>
<td>$P_1$</td>
<td>$c_1$</td>
</tr>
<tr>
<td></td>
<td>$S_2$</td>
<td>$P_2$</td>
<td>$c_2$</td>
</tr>
<tr>
<td></td>
<td>$S_3$</td>
<td>$P_3$</td>
<td>$c_3$</td>
</tr>
<tr>
<td></td>
<td>$S_4$</td>
<td>$P_4$</td>
<td>$c_4$</td>
</tr>
<tr>
<td></td>
<td>$S_5$</td>
<td>$P_5$</td>
<td>$c_5$</td>
</tr>
<tr>
<td></td>
<td>$S_6$</td>
<td>$P_6$</td>
<td>$c_6$</td>
</tr>
<tr>
<td>Node 2</td>
<td>$S_{M-1}$</td>
<td>$P_{M-1}$</td>
<td>$c_{M-1}$</td>
</tr>
<tr>
<td>Node K</td>
<td>$S_{M}$</td>
<td>$P_{M}$</td>
<td>$c_{M}$</td>
</tr>
</tbody>
</table>

One bit data from sensor node $k$th

Figure 2.2: Sub-carrier and Power allocation in WSN.

missions and thus having the lowest error probability). Consequently, being the highest quality transfer, this is the most expensive one, meanwhile the second due to the limited interference has a higher error probability but lower cost.

The problem is that nodes $U_k, \{k = 1, ..., K\}$ need to transmit an $Y_k$ amount of packets to the BS with minimal cost to guarantee that at least $X_k$ amount of packets arrive with a given probability $(1 − \gamma)$. This entails a redundant packets’ transmission $Y_k > X_k$, assuming that the information content of the packets is redundant (i.e., packets with the same information are available in abundance e.g., having the same frames in the case of video information). Hence, the problem is characterized by following parameters: $(X_k, K_f, \gamma)$, where $X_k$ is the number of packets to be transmitted by node $U_k$, $K_f$ is the number of available carrier frequencies in a sub-carrier, and $\gamma$ is the reliability parameter related to the probability of transmitting at least $X_k$ packets to the BS.

There are different quality channels are available for transmission, which are characterized by their error probabilities (probability of failed packets at each sub-carrier) denoted by $\hat{p} = (p_1, ..., p_M)$. The corresponding cost of a channel in the different sub-carriers are characterized by vector $\hat{c} = (c_1, ..., c_M)$. The cost is some monotone decreasing function of the failure probability e.g., $c_i = \Psi(1/p_i)$.

In order to send $X_k$ packets with a given reliability via unreliable channels, a node $U_k$ may choose to send $Y_k$ packets, where $Y_k > X_k$. The packets assignment to the different sub-carriers are represented by a vector $\vec{y} = (y_{1k}, ..., y_{Mk})$, where $y_{ik}$ indicates the amount of packets allocated to sub-carrier $S_i$ by node $U_k$ with:

$$\sum_{m=1}^{M} y_{mk} = Y_k. \quad (2.8)$$

The number of packets transmitted in the different sub-carriers by node $U_k$ are $y_{mk}, \{m = 1, ..., M\}$. Due to the unreliability of the channels let $\xi_{mk}$ be the random variable indicating the amount of packets were sent in sub-carrier $S_m$, and came to the BS successfully
The reliability constraint is then defined as:

\[ P \left( \sum_{m=1}^{M} \xi_{mk} \geq X_k \right) \geq 1 - \gamma. \]  \hfill (2.9)

Where the probability of the amount of packets sent \( u_{mk} \) in sub-carrier \( m^{th} \) is calculated as in (2.10)

\[ P(\xi_{mk} = u_{mk}) = \left( \frac{y_{mk}}{u_{mk}} \right) (1 - p_m)^{u_{mk}} p_m^{(y_{mk} - u_{mk})}. \]  \hfill (2.10)

Furthermore, I assume the following ordering \( c_1 \leq c_2 \leq \ldots \leq c_M \) and consequently \( \kappa_1 \leq \kappa_2 \leq \ldots \leq \kappa_M \) where \( \kappa_m = 1 - P_m, \{m = 1, \ldots, M\} \).

---

**Figure 2.3:** Illustration of data transmission between node \( k^{th} \) and the BS

Figure 2.3 depicts the data transmission between a sensor node and the Base Station. After sensing the environmental parameters, sensor node \( k^{th} \) will schedule the number of packets needed to be sent to the BS into different sub-carriers to yield a minimal cost packet allocation. The cost of sending data packets from sensor node \( k^{th} \) to the BS is calculated by:

\[ C(k) = \sum_{i=1}^{M} c_i y_{ik}. \]  \hfill (2.11)

Now, one can formulate the optimal scheduling problem as finding the optimal vector \( \mathbf{y}_{opt} \) for which:
Objective:

\[ y_{opt} : \min_y \sum_{k=1}^{K} \sum_{m=1}^{M} y_{mk} c_m. \]  

subject to the constraints:

\[ C_1 : P \left( \sum_{k=1}^{K} \sum_{m=1}^{M} \xi_{mk} \geq X_k \right) \geq 1 - \gamma \]
\[ C_2 : \max_i y_{mk} \leq K_f \]

2.3 Solution by random scheduling

The optimization problem in (2.12) is generally very hard to solve. The optimal solution can be found by computing all possible allocation cases, however, this is clearly beyond the capacity of the available resources as this technique will require \( O(M^{Y_k}) \) ways for assigning \( Y_k \) packets needed to be sent from \( M \) sub-carriers [132]. One must also calculate the probability of at least \( X_k \) packets reaching the BS in order to choose among those allocations the best one with respect to cost.

To solve this problem, in a suboptimal manner, a very simple sub-carrier allocation was proposed in [15]. This algorithm is based on random scheduling (RS), which assigns a number of packets to each sub-carrier randomly. The best scheduler will be chosen by
Monte Carlo technique, and its procedure is described in detail by the following flowchart (Figure 2.4). The steps of the RS algorithm are fully described in algorithm 1.

**Algorithm 1: random scheduling algorithm**

1. **Input:** Parameters of model \((X_k, K_f, \gamma), M;\)

2. **Output:** Optimized cost schedule \(y_{\text{opt}} : \min_y \sum_{k=1}^{K} \sum_{m=1}^{M} y_{mk}c_m;\)

3. **Step 0:** Initialization

4. Set the initial model \((X_k, K_f, \gamma), M,\) the best system cost \(C_{\text{min}},\) the probability of failure \(\bar{\kappa} = (\kappa_1, \kappa_2, \ldots, \kappa_M)\) with \(0 \leq \kappa_1 \leq \kappa_2 \leq \ldots \leq \kappa_M \leq 1\) and the price in each sub-carrier

5. \(\bar{c} = (c_1, c_2, \ldots, c_M)\) with \(\sum_{i=1}^{M} c_i = 1\)

6. **Step 1:** Generate vector \(y = (y_{1k}, \ldots, y_{Mk})\) randomly with \(\max_{m} y_{mk} \leq K\) and \(\sum_{m=1}^{M} y_{mk} = Y_k \geq X_k\)

7. **Step 2:** Calculate the system reliability \(P(t);\)

8. **Step 3:** Check the system reliability is better than a predefined threshold as in (2.2). If not, go back to step 1:

9. **Step 4:** If yes, get the system cost for schedule of \(t^{th}: C(t) = \bar{y}(t)\bar{c};\)

10. **Step 5:** Check whether the newly obtained solution \(C(t)\) is cheaper than the previous system cost \(C_{\text{min}};\)

11. **Step 6:** If yes, achieve new optimal schedule \(\bar{y}_{\text{opt}}\) and the best system cost \(C_{\text{min}} = C(t).\) Other wise, increase the number of simulation \(t = t + 1,\) and then go to step 7;

12. **Step 7:** Check if \(t\) equals to the simulation parameter \(N_{\text{simpara}}\) then the algorithm ends. If not, go back to step 1.

Unfortunately, in the case of running the RS algorithm, we need to increase the value of simulation parameter \((N_{\text{simparam}})\) to be big enough to get the optimal scheduling \(\bar{y}_{\text{opt}}.\) It is one of the main reasons for the long running time of the RS algorithm. To avoid the long running time, in the next section, I propose an algorithm for allocating \(Y\) packets to \(M\) sub-carriers with minimum cost, in the shortest running time and with a predefined system reliability.

### 2.4 The proposed algorithm

Let us assume that we have a packet assignment scheme specified by vector \(y_{1 \times M} = [y_{1k}, \ldots, y_{Hk}, 0, \ldots, 0].\) The initial values of the system parameters are given in Table 2.1.
Table 2.1: The specification of a packet assignment scheme

<table>
<thead>
<tr>
<th>Sub-carriers</th>
<th>S₁</th>
<th>S₂</th>
<th>...</th>
<th>S₇</th>
<th>S₈</th>
<th>...</th>
<th>S₉</th>
</tr>
</thead>
<tbody>
<tr>
<td>The correponding cost</td>
<td>c₁</td>
<td>c₂</td>
<td>...</td>
<td>c₇</td>
<td>c₈</td>
<td>...</td>
<td>c₉</td>
</tr>
<tr>
<td>Probability of success</td>
<td>κ₁</td>
<td>κ₂</td>
<td>...</td>
<td>κ₇</td>
<td>κ₈</td>
<td>...</td>
<td>κ₉</td>
</tr>
<tr>
<td># of packets sent</td>
<td>y₁ₖ</td>
<td>y₂ₖ</td>
<td>...</td>
<td>y₇ₖ</td>
<td>y₈ₖ</td>
<td>0</td>
<td>...</td>
</tr>
</tbody>
</table>

The probability of all the packets sent without error is given as:

\[ p(y) = \prod_{m=1}^{M} \kappa_{m}^{y_{mk}}. \]  \hspace{1cm} (2.13)

Table 2.2: Moving a packet from sub-carrier S₇ to sub-carrier S₈

<table>
<thead>
<tr>
<th>From</th>
<th>Sub-carriers</th>
<th>S₁</th>
<th>...</th>
<th>S₇</th>
<th>S₈</th>
<th>...</th>
<th>S₉</th>
</tr>
</thead>
<tbody>
<tr>
<td>The correponding cost</td>
<td>c₁</td>
<td>c₂</td>
<td>...</td>
<td>c₇</td>
<td>c₈</td>
<td>...</td>
<td>c₉</td>
</tr>
<tr>
<td>Probability of success</td>
<td>κ₁</td>
<td>κ₂</td>
<td>...</td>
<td>κ₇</td>
<td>κ₈</td>
<td>...</td>
<td>κ₉</td>
</tr>
<tr>
<td># of packets sent</td>
<td>y₁ₖ</td>
<td>y₂ₖ</td>
<td>...</td>
<td>y₇ₖ</td>
<td>y₈ₖ</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>The system cost</td>
<td>c₁ y₁ₖ</td>
<td>c₂ y₂ₖ</td>
<td>...</td>
<td>c₇ y₇ₖ</td>
<td>c₈ y₈ₖ</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>The system reliability</td>
<td>(κ₁)⁺¹</td>
<td>(κ₂)⁺¹</td>
<td>...</td>
<td>(κ₇)⁺¹</td>
<td>(κ₈)⁺¹</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To</th>
<th>Sub-carriers</th>
<th>S₁</th>
<th>...</th>
<th>S₇</th>
<th>S₈</th>
<th>...</th>
<th>S₉</th>
</tr>
</thead>
<tbody>
<tr>
<td>The correponding cost</td>
<td>c₁</td>
<td>c₂</td>
<td>...</td>
<td>c₇</td>
<td>c₈</td>
<td>...</td>
<td>c₉</td>
</tr>
<tr>
<td>Probability of success</td>
<td>κ₁</td>
<td>κ₂</td>
<td>...</td>
<td>κ₇</td>
<td>κ₈</td>
<td>...</td>
<td>κ₉</td>
</tr>
<tr>
<td># of packets sent</td>
<td>y₁ₖ</td>
<td>y₂ₖ</td>
<td>...</td>
<td>y₇ₖ</td>
<td>y₈ₖ⁻¹</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>The system cost</td>
<td>c₁ y₁ₖ</td>
<td>c₂ y₂ₖ</td>
<td>...</td>
<td>c₇ y₇ₖ⁻¹</td>
<td>c₈ y₈ₖ⁻¹</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>The system reliability</td>
<td>(κ₁)⁺¹</td>
<td>(κ₂)⁺¹</td>
<td>...</td>
<td>(κ₇)⁺¹</td>
<td>(κ₈)⁺¹⁻¹</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

Property:
The system reliability will be increased (i.e., the value of probability of all the packets sent without error) in two ways: either (i) moving a packet from sub-carrier S₇ to sub-carrier S₈, \{H = 1, ..., M - 1\}; or (ii) sending a new packet in sub-carrier S₁. Here I assume that both of these changes can be issued in terms of either in sub-carrier S₈, or in sub-carrier S₁ there are no more packets to send than \(K_f - 1\).

Proof

- **Evaluating the change in reliability when moving a packet from sub-carrier S₇ to sub-carrier S₈**
  
  This type of state change occurs when the packet assignment vector \(y(t) = [y₁ₖ, ..., y₇ₖ, 0, ..., 0]\) changes to \(y(t + 1) = [y₁ₖ, ..., y₇ₖ, -1, 1, 1, 0, ..., 0]\). The process of moving a packet from sub-carrier S₇ to sub-carrier S₈ is described detail in the Table 2.2.
The change of the cost function can be expressed as:

\[ C(t) = \sum_{i=1}^{H} c_i y_{ik} \]  

(2.14)

and

\[ C(t + 1) = \sum_{i=1}^{H-1} c_i y_{ik} + c_H (y_{Hk} - 1) + c_{H+1} \]

\[ = C(t) + c_{H+1} - c_H. \]  

(2.15)

As a result

\[ \Delta C(t) = C(t + 1) - C(t) = c_{H+1} - c_H. \]  

(2.16)

The corresponding reliability changes from

\[ p(y(t)) = \prod_{m=1}^{M} \kappa_{ymk} \]  

(2.17)

to

\[ p(y(t + 1)) = \frac{\kappa_{H+1}}{\kappa_H} \prod_{m=1}^{M} \kappa_{ymk}. \]  

(2.18)

One may see that indeed the reliability has been increased by this state change.

- **Evaluating the change in reliability when sending a new packet in sub-carrier \( S_1 \).**

  In this case the state vector changes from \( y(t) = [y_{1k}, y_{2k}, ..., y_{Hk}, 0...0] \) to \( y(t + 1) = [y_{1k} + 1, y_{2k}, ..., y_{Hk}, 0...0] \) and these changes are given in the Table 2.3. The change in reliability and cost function are evaluated for each sub-carrier.

### Table 2.3: Sending a new packet in sub-carrier \( S_1 \)

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-carriers</td>
<td>( S_1 )</td>
</tr>
<tr>
<td>The corresponding cost</td>
<td>( c_1 )</td>
</tr>
<tr>
<td>The probability of success</td>
<td>( \kappa_1 )</td>
</tr>
<tr>
<td># of packets sent</td>
<td>( y_{1k} )</td>
</tr>
<tr>
<td>The system cost</td>
<td>( c_1 y_{1k} )</td>
</tr>
<tr>
<td>The system reliability</td>
<td>( (\kappa_1)^{y_{1k}} )</td>
</tr>
</tbody>
</table>

| Sub-carriers | \( S_1 \) | \( S_2 \) | ... | \( S_H \) | \( S_{H+1} \) | ... | \( S_M \) |
| The corresponding cost | \( c_1 \) | \( c_2 \) | ... | \( c_H \) | \( c_{H+1} \) | ... | \( c_M \) |
| The probability of success | \( \kappa_1 \) | \( \kappa_2 \) | ... | \( \kappa_H \) | \( \kappa_{H+1} \) | ... | \( \kappa_M \) |
| # of packets sent | \( y_{1k} + 1 \) | \( y_{2k} \) | ... | \( y_{Hk} \) | 0 | ... | 0 |
| The system cost | \( c_1 (y_{1k} + 1) \) | \( c_2 y_{2k} \) | ... | \( c_M y_{Hk} \) | 0 | ... | 0 |
| The system reliability | \( (\kappa_1)^{(y_{1k} + 1)} \) | \( (\kappa_2)^{y_{2k}} \) | ... | \( (\kappa_H)^{y_{Hk}} \) | 1 | ... | 1 |
In cost is from

\[ C(t) = \sum_{i=1}^{M} c_i y_{ik} \]  
(2.19)

to

\[ C(t + 1) = \sum_{i=2}^{H} c_i y_{ik} + c_1 (y_{Hk} + 1) = C(t) + c_1. \]  
(2.20)

As a result

\[ \Delta C(t) = C(t + 1) - C(t) = c_1. \]  
(2.21)

The corresponding change in reliability can be calculated as

\[ p(t + 1) = \kappa_1 p(t) + (1 - \kappa_1) p(t) + \kappa_1 p(t - 1) \]

which yields

\[ p(t + 1) = p(t) + \kappa_1 p(t - 1). \]  
(2.22)

As a conclusion, one can state that if we move or send a new packet the reliability parameter will increase in both cases. The cost also increases, but both transformations are defined in such a way that they entail minimal cost increases among moving a packet or sending a new packet. Based on these facts, one can devise the optimal algorithm given as follows.

Initial state: Assign the packets for transfer as follows

\[ X = H \cdot K + t \] then \( y_i = K, i = 1, \ldots, H \) and \( y_{H+1} := m \). Then check the condition

\[ p(y(t)) = \prod_{i=1}^{M} \kappa_i y_{ik} \geq 1 - \gamma \]

if yes, the algorithm will stop, if not, make a move or send a new packet depending on the cost change which is smaller. Check the reliability condition when arriving at a new schedule vector by calculating either

\[ p(y(t+1)) = \frac{\kappa_{H+1}}{\kappa_H} \prod_{i=1}^{M} \kappa_i y_{ik} \]

or

\[ p(t + 1) = p(t) + \kappa_1 p(k - 1). \]

Terminate the algorithm if \( p(y(t)) \geq 1 - \gamma \) is met. The steps of the optimized cost scheduling algorithm are given by algorithm 2.

2.5 Numerical results

In this section, I investigate the performance of the proposed resource allocation scheme compared to the RS algorithm. Both algorithms are coded by Matlab R2017a and executed on a 2.4 GHz PC with 8 GB RAM.

2.5.1 Experimental scenarios

In order to compare the performance of the proposed CMS algorithm with other algorithms, a large simulation dataset is created. I conducted some experimental studies with the different number of sub-carriers and the number of packets needed to be sent. I set the value of \((X,M)\) to be equal to \((25, 10), (25, 15), (25, 20), (100, 50),\) and \((100, 100),\) respectively. These changes are summarized in following scenarios, where the criteria for performance are considered such as the system cost, the system reliability and the running time of the CMS algorithm and the RS algorithm.
Algorithm 2: optimized cost scheduling algorithm

1 **Input:** Parameters of model \((X_k, K, \gamma), M;\) 

2 **Output:** Optimized cost schedule \(y_{\text{opt}} : \min \sum_{m=1}^{M} y_{mk}c_{m};\) 

3 **Step 0: Initialization** 

4 Set the initial model \((X_k, K, \gamma),\) number of sub-carrier \(M,\) the probability of failure \(\bar{\kappa} = (\kappa_1, \kappa_2, ..., \kappa_M)\) with \((0 \leq \kappa_1 \leq \kappa_2 \leq ... \leq \kappa_M \leq 1)\) and the cost \(c = (c_1, c_2, ..., c_M)\) with \(\sum_{i=1}^{M} c_i = 1.\) 

5 Set the initial schedule \(y_{\text{initial}} = (y_{1k} = Y, y_{2k} = 0, ..., y_{Mk} = 0);\) 

7 **Step 1:** Check the condition \(P\left(\sum_{m=1}^{M} \xi_{mk} \geq X_k\right) \geq 1 - \gamma\) if yes, go to Step 6 and not go to Step 2. 

8 **Step 2:** Calculate: 

\[
\begin{cases} 
\Delta C_1(t) = C(t + 1) - C(t) \\
\Delta C_2(t) = C(t + 1) - C(t)
\end{cases} \iff \begin{cases} 
\Delta C_1(t) = c_{H+1} - c_H \\
\Delta C_2(t) = c_1
\end{cases}
\]

10 **Step 3:** Check the condition \(\Delta C_1(t) \leq \Delta C_2(t)\) if yes, go to Step 4 and not go to Step 5. 

11 **Step 4:** Move a packet from sub-carrier \(S_H\) to sub-carrier \(S_{H+1}.\) \((y_{Hk} = -; y_{(H+1)k} = +)\) After moving a packet, The algorithm will go back to Step 1. 

12 **Step 5:** Send a new packet in sub-carrier \(S_1.\) 

13 \((y_{1k} = +; Y = +)\) and then the algorithm will continue from Step 1. 

14 **Step 6:** Algorithm exit with optimized cost schedule vector: \(y_{\text{opt}} = [y_{1k}, y_{2k}, ..., y_{Mk}].\) 

2.5.2 Comparative performance analysis

To evaluate the efficiency of the proposed CMS algorithm, I focus on comparing the system cost and running time of the two algorithms CMS and RS, respectively. I simulated the proposed algorithms under the number of packets needed to be sent is \(X = 25\) by using \(M = 20\) sub-carriers and the required system reliability is \(0.99 \times \gamma = 0.1.\) The two important performance indicators used to measure the efficiency of the algorithms are the system cost and the elapsed time. They are described in Figure 2.5 and 2.6, respectively. 

In Figure 2.5, we can see that the maximum system cost of my algorithm is less than 0.8 (cost units) while the minimum system cost of the RS algorithm is higher than 0.9 (cost units). The average of system cost by the RS algorithm is always higher from 2 to 10 times more than that of the CMS algorithm. It means that my algorithm can save at least 50% of energy thus it prolongs the network lifetime. 

Figure 2.6 shows the elapsed time of two algorithms with the same initial parameters. In Figure 2.6a, one can observe that the CMS algorithm gives the results faster than the RS
algorithm. The histogram of the elapsed time of both algorithms are described in Figure 2.6b. With the higher frequency in the range of small elapsed time, the CMS algorithm spent not much time to find the optimal scheduling meanwhile the RS algorithm need to increase the value of parameter \( N_{simparam} \) to big enough in order to get the optimal scheduling so the RS algorithm runs more slowly than the CMS algorithm.

Figure 2.5: The comparison of system cost

Figure 2.6: The elapsed time of algorithms
The comparison of the system cost under the system reliability constraint:

In order to measure the improvement in system cost of two algorithms, I use the percentage improvement in the price paid for sending data as given in Equation (2.23).

\[ P_{\text{improve}} = \left( 1 - \frac{\text{Cost}_{\text{CMS}}}{\text{Cost}_{\text{RS}}} \right) \times 100. \]  

(2.23)

where Cost\text{CMS} and Cost\text{RS} are the system cost of the CMS algorithm and the RS algorithm, respectively. At the first scenario, the model with \( X = 25 \) packets needed to be sent from number of sub-carriers is \( M = 10, 15, \) and \( 20 \) while the system reliability required is 0.99. The results of these scenarios are given in Table 2.4, 2.5, 2.6, and described in Figures 2.7a, 2.7b and 2.7c, respectively.

### Table 2.4: The improvement in system cost with \( M = 10 \)

<table>
<thead>
<tr>
<th># of packets sent</th>
<th>System reliability</th>
<th>The system cost</th>
<th>( P_{\text{improve}} ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RS</td>
<td>CMS</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>0.62</td>
<td>0.214</td>
<td>0.195</td>
</tr>
<tr>
<td></td>
<td>0.65</td>
<td>1.049</td>
<td>0.976</td>
</tr>
<tr>
<td></td>
<td>0.66</td>
<td>1.292</td>
<td>1.181</td>
</tr>
<tr>
<td>30</td>
<td>0.64</td>
<td>0.317</td>
<td>0.298</td>
</tr>
<tr>
<td></td>
<td>0.84</td>
<td>1.302</td>
<td>1.204</td>
</tr>
<tr>
<td></td>
<td>0.96</td>
<td>1.866</td>
<td>1.736</td>
</tr>
<tr>
<td>35</td>
<td>0.64</td>
<td>0.742</td>
<td>0.678</td>
</tr>
<tr>
<td></td>
<td>0.84</td>
<td>2.208</td>
<td>1.996</td>
</tr>
<tr>
<td></td>
<td>0.99</td>
<td>2.812</td>
<td>2.663</td>
</tr>
</tbody>
</table>

The improvement in system cost: 7.6 %

### Table 2.5: The improvement in system cost with \( M = 15 \)

<table>
<thead>
<tr>
<th># of packets sent</th>
<th>System reliability</th>
<th>The system cost</th>
<th>( P_{\text{improve}} ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RS</td>
<td>CMS</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>0.66</td>
<td>0.487</td>
<td>0.452</td>
</tr>
<tr>
<td></td>
<td>0.68</td>
<td>0.931</td>
<td>0.598</td>
</tr>
<tr>
<td></td>
<td>0.69</td>
<td>0.946</td>
<td>0.9</td>
</tr>
<tr>
<td>30</td>
<td>0.66</td>
<td>0.549</td>
<td>0.543</td>
</tr>
<tr>
<td></td>
<td>0.84</td>
<td>0.935</td>
<td>0.854</td>
</tr>
<tr>
<td></td>
<td>0.98</td>
<td>1.179</td>
<td>1.108</td>
</tr>
<tr>
<td>35</td>
<td>0.66</td>
<td>0.65</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>0.84</td>
<td>0.954</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>0.99</td>
<td>1.391</td>
<td>1.243</td>
</tr>
</tbody>
</table>

The improvement in system cost: 9.8 %

In order to improve the quality of data transmission, in the next scenarios, I evaluate the performance of the proposed algorithm when increasing the number of packets needs to be sent is \( X = 100 \), the number of sub-carriers in source node is \( M = 50 \) and \( M = 100 \). The improvements in the system cost are given in Table 2.7 and 2.8. The detailed comparison of the system cost between the RS algorithm and the CMS algorithm are described in Figure 2.7d and 2.7e, respectively.
<table>
<thead>
<tr>
<th>L = 10; Y = 25</th>
<th>L = 10; Y = 30</th>
<th>L = 10; Y = 35</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS Algorithm</td>
<td>CMS Algorithm</td>
<td></td>
</tr>
<tr>
<td>0.314</td>
<td>0.317</td>
<td>0.376</td>
</tr>
<tr>
<td>1.049</td>
<td>1.080</td>
<td>1.307</td>
</tr>
<tr>
<td>2.663</td>
<td>2.703</td>
<td>2.812</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L = 15; Y = 25</th>
<th>L = 15; Y = 30</th>
<th>L = 15; Y = 35</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS Algorithm</td>
<td>CMS Algorithm</td>
<td></td>
</tr>
<tr>
<td>0.467</td>
<td>0.468</td>
<td>0.479</td>
</tr>
<tr>
<td>0.935</td>
<td>0.936</td>
<td>0.954</td>
</tr>
<tr>
<td>2.208</td>
<td>2.228</td>
<td>2.384</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L = 20; Y = 25</th>
<th>L = 20; Y = 30</th>
<th>L = 20; Y = 35</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS Algorithm</td>
<td>CMS Algorithm</td>
<td></td>
</tr>
<tr>
<td>0.491</td>
<td>0.491</td>
<td>0.491</td>
</tr>
<tr>
<td>1.179</td>
<td>1.182</td>
<td>1.244</td>
</tr>
<tr>
<td>2.812</td>
<td>2.813</td>
<td>2.873</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L = 50; Y = 100</th>
<th>L = 50; Y = 103</th>
<th>L = 50; Y = 105</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS Algorithm</td>
<td>CMS Algorithm</td>
<td></td>
</tr>
<tr>
<td>0.467</td>
<td>0.468</td>
<td>0.479</td>
</tr>
<tr>
<td>1.179</td>
<td>1.182</td>
<td>1.244</td>
</tr>
<tr>
<td>2.812</td>
<td>2.813</td>
<td>2.873</td>
</tr>
</tbody>
</table>

Figure 2.7: The improvement in system cost
### Table 2.6: The improvement in system cost with $M = 20$

<table>
<thead>
<tr>
<th># of packets sent</th>
<th>System reliability</th>
<th>The system cost</th>
<th>$P_{\text{improve}}$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RS</td>
<td>CMS</td>
</tr>
<tr>
<td>25</td>
<td>0.63</td>
<td>0.25</td>
<td>0.23</td>
</tr>
<tr>
<td></td>
<td>0.64</td>
<td>0.41</td>
<td>0.369</td>
</tr>
<tr>
<td></td>
<td>0.65</td>
<td>0.516</td>
<td>0.489</td>
</tr>
<tr>
<td>30</td>
<td>0.66</td>
<td>0.634</td>
<td>0.589</td>
</tr>
<tr>
<td></td>
<td>0.84</td>
<td>0.826</td>
<td>0.741</td>
</tr>
<tr>
<td></td>
<td>0.98</td>
<td>1.56</td>
<td>1.357</td>
</tr>
<tr>
<td>35</td>
<td>0.66</td>
<td>0.89</td>
<td>0.775</td>
</tr>
<tr>
<td></td>
<td>0.84</td>
<td>1.469</td>
<td>1.244</td>
</tr>
<tr>
<td></td>
<td>0.99</td>
<td>2.626</td>
<td>2.384</td>
</tr>
</tbody>
</table>

The improvement in system cost: **10.12 %**

### Table 2.7: The improvement in system cost with $M = 50$

<table>
<thead>
<tr>
<th># of packets sent</th>
<th>System reliability</th>
<th>The system cost</th>
<th>$P_{\text{improve}}$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RS</td>
<td>CMS</td>
</tr>
<tr>
<td>100</td>
<td>0.72</td>
<td>3.27</td>
<td>2.15</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>3.65</td>
<td>2.68</td>
</tr>
<tr>
<td></td>
<td>0.78</td>
<td>3.82</td>
<td>3.42</td>
</tr>
<tr>
<td>103</td>
<td>0.75</td>
<td>3.7</td>
<td>2.75</td>
</tr>
<tr>
<td></td>
<td>0.86</td>
<td>4.21</td>
<td>3.84</td>
</tr>
<tr>
<td></td>
<td>0.98</td>
<td>4.98</td>
<td>4.56</td>
</tr>
<tr>
<td>105</td>
<td>0.75</td>
<td>3.86</td>
<td>2.85</td>
</tr>
<tr>
<td></td>
<td>0.88</td>
<td>4.94</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>0.99</td>
<td>5.8</td>
<td>5.18</td>
</tr>
</tbody>
</table>

The improvement in system cost: **17.78 %**

### Table 2.8: The improvement in system cost with $M = 100$

<table>
<thead>
<tr>
<th># of packets sent</th>
<th>System reliability</th>
<th>The system cost</th>
<th>$P_{\text{improve}}$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RS</td>
<td>CMS</td>
</tr>
<tr>
<td>100</td>
<td>0.81</td>
<td>5.21</td>
<td>4.31</td>
</tr>
<tr>
<td></td>
<td>0.92</td>
<td>6.38</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>0.96</td>
<td>6.5</td>
<td>5.42</td>
</tr>
<tr>
<td>102</td>
<td>0.85</td>
<td>5.84</td>
<td>4.62</td>
</tr>
<tr>
<td></td>
<td>0.95</td>
<td>6.49</td>
<td>4.85</td>
</tr>
<tr>
<td></td>
<td>0.98</td>
<td>6.92</td>
<td>5.62</td>
</tr>
<tr>
<td>103</td>
<td>0.86</td>
<td>5.96</td>
<td>4.76</td>
</tr>
<tr>
<td></td>
<td>0.98</td>
<td>7.32</td>
<td>5.82</td>
</tr>
<tr>
<td></td>
<td>0.99</td>
<td>8.61</td>
<td>6.82</td>
</tr>
</tbody>
</table>

The improvement in system cost: **21.08 %**

As we can observe from Tables 2.4, 2.5, 2.6, 2.7 and Table 2.8 of 5 scenarios when we increase the number of sub-carrier from $M = 10$ to $M = 100$, the system costs decreased down to 7.6%, 9.8%, 10.12%, 17.18% and 21.08% respectively among the CMS algorithm and the RS algorithm. Hence, the proposed algorithm decreases the system cost of data transmission with the same expected system reliability.
Improving the reliability of the data transmission:

In order to reach the reliable data transfer, each sensor node has to send the number of packets $Y_k$ being bigger than the original data size $X_k$. The algorithm must obtain a given level of reliability with as less data redundancy as possible. In this section, with the system $(X, M) = (25, 10)$, we will compare the number of packets which node in the RS algorithm and the CMS algorithm sent for the same expected transmission system reliability (0.999). The results are described in Figure 2.8. It can be seen that the node in the RS algorithm need to send much more data packets than the CMS algorithm does in order to reach the system reliability (0.999).

**Figure 2.8:** Improving the reliability of data transmission with $(X, M) = (25, 10)$

The comparison of the data rate versus the number of nodes:

Figure 2.9 depicts the performance in data rate between my proposal and some typical algorithms. It is shown that with a higher number of nodes, the total data rates will increase. The reason for this problem is in the multiuser diversity if the number of nodes in the system increases, the probability of sharing the same resources becomes lower. In Figure (2.9), it is shown that the sum of the nodes data rates achieved by a static TDMA system, in which each node is allocated with the equal transmission power, remained flat for all 8 nodes, while in other algorithms, they grew together with the number of nodes. Although the node data rates of the CMS algorithm are slightly lower than that of the algorithm in [70], they still are better performances than algorithms in [61,118] and static TDMA.
The comparison of the fairness pointer versus the number of nodes:

In this section, my proposed CMS algorithm is compared with the algorithms proposed in [61, 70, 118] under fairness criterion. The fairness criterion is introduced in [139] and given in Equation (2.24).

\[ F = \frac{\left( \sum_{k=1}^{K} R_k \right)^2}{K \sum_{k=1}^{K} (R_k)^2} \]  

(2.24)

The Fairness pointer \( F \) is a measurement parameter of the equal data rates among nodes in the system, and its values range from 0 to 1. It is better for any resource allocation algorithm if it has a higher value of the fairness pointer. It can be seen in Figure 2.10 that the algorithms in [61, 70] have significant unequal among the nodes in the system. The fairness pointers of these algorithms are low level and decrease with increasing the number of nodes. While the static TDMA scheme supports a better fairness pointer but it is still much lower than my achieved results and the outcome of the algorithm in [118]. Thus my algorithm can improve the fairness among the nodes in the system.
2.6 Conclusions and directions for future research

In sensing applications with a large amount of data to be sent over the WSN with the limited resources (e.g., energy capacity, memory usage, and network bandwidth), low-cost packet transfer is an imperative requirement for wireless transmission. In this chapter, I presented a new efficient algorithm to transmit data with low energy consumption by developing a smart scheduler. This algorithm ensures a predefined level of reliability with minimum data redundancy. The numerical results demonstrate that my CMS algorithm achieves better performances in data transmission than some typical algorithms. It is noteworthy that my algorithm does not only minimize the system cost but also decreases the running time. In this way, WSN is capable of capturing the real-time events. The results reported vindicate that the newly proposed CMS algorithm can indeed be applied in mission-critical WSNs. It could be developed to further enhance monitoring performance, and they could be analyzed with the introduction of short data transmission time, high reliability, and high data-rate communication.
Chapter 3

An Efficient Approach for Maximizing Lifespan in Wireless Sensor Networks by Using Mobile Sinks

Recently, sink mobility has been shown to be highly beneficial in improving network lifetime in wireless sensor networks. Numerous studies have exploited mobile sinks to collect sensed data in order to improve the energy efficiency and reduce the WSN operational costs. However, there have been few studies on the effectiveness of MS operation on WSN closed operating cycles. Therefore, it is important to investigate how data is collected and how to plan the trajectory of the MS in order to gather data in time, reduce energy consumption, and improve the network lifetime. In this chapter, I combine two methods (e.g., the cluster-head election algorithm and the MS trajectory optimization algorithm), to propose the optimal MS movement strategy. This chapter aims to provide a closed operating cycle for WSNs, by which the energy consumption and running time of a WSN is minimized during the cluster election and data gathering periods. Furthermore, my flexible MS movement scenarios achieve both long network lifetime and short execution time. The simulation results demonstrate that my proposed algorithms achieve better performance than other well-known algorithms.

3.1 Introduction

In recent years, many authors have focused their investigations on improving the wireless sensor network lifetime by mobility-based technology. Some studies [5, 86] tried to change the initial deployment of the network by mobile nodes. Some other approaches [39, 105, 148] utilized MSs to collect data in order to enhance the sensing coverage, connectivity, and network lifetime as well. However, there has been little research work on WSN closed operating cycles to find an optimal solution for improving the network lifetime. Therefore, in this chapter, I focus on extending the network lifetime by improving system energy efficiency in a closed cycle, including cluster head election, data collection, and data transmission.

It is known that in order to achieve energy efficient management, clustering is a promising technique by which all Cluster member Nodes (CNs) transmit their sensing data to a CH
before forwarding them to the MS. Furthermore, the energy resources and traveling time of a MS can also be limited. Therefore, it needs energy-efficient technologies in order to collect the sensing data with minimum energy consumption, and within the predefined time deadline. In this chapter, I also consider several problems of mobility-based technology, such as where, when, and how to obtain sensing data while minimizing sensor node energy consumption in a given limited time.

It is proved that a CH will consume much more energy than the normal nodes. Therefore, the cluster-head election problem is highly critical for balancing the energy consumption among the nodes in the network to increase the network lifetime.

Extended research has been carried out on CH election in WSNs [25,58] in order to reduce energy consumption and prolong the network lifetime. Low-energy adaptive clustering hierarchy (LEACH) [50] is one of these well-known clustering protocols. The key technology of LEACH is randomized rotation of local CHs to distribute energy among the nodes in the network. One normal node \((n)\) may be a CH in the recent \((t \mod (1/p))\) rounds with probability \(P\):

\[
P(n) = \begin{cases} 
p & \text{if } n \in G \\
0 & \text{otherwise} \end{cases}
\]

where \(p\) is the desired percentage of CHs, \(t\) denotes the current round, and \(Q\) indicates the set of nodes that have not been a CH in the last \((1/p)\) rounds. As indicated in [75], by LEACH algorithm, the energy consumption is reduced significantly by using a Time Division Multiple Access (TDMA) schedule. Unfortunately, this clustering protocol still has several shortcomings: (i) a single hop directly between a CH and the Base Station (BS) will be impossible with a large sensing field, and it may dissipate much energy for data transmission; (ii) it is unreasonable for load balancing when changing the CHs if it does not take into account the residual energy and geographic location of each sensor node; and (iii) it is not certain that CHs will be uniformly distributed throughout the network when a normal node is chosen to be a CH on the basis of a random number. Therefore, energy dissipation becomes inefficient. In order to overcome these shortcomings, some improved versions of LEACH have been proposed. The Impro-LEACH algorithm in [165] is one example that can improve the network lifetime by modifying the LEACH threshold formula. However, it still compares a random number with the probability to elect CHs as proposed in LEACH algorithm. Therefore, the method proposed in [165] does not overcome the drawbacks of the LEACH method. Moreover, these methods are still essentially designed for CH election and do not take into account the combination of energy consumption in each sensor node and the energy balance expenditure among nodes in the network. To eliminate this problem, I propose an efficient algorithm for choosing: (i) the number of clusters in a network, which guarantees energy balance among nodes; (ii) the best location of a single CH for each cluster, where the CH not only receives data from all CNs with the minimum energy consumption but also guarantees the energy balance among nodes in the network.

I also consider scenarios related to MS trajectory in mobile wireless sensor networks (MWSNs). Herein, an optimal trajectory of the MS is obtained when both minimizing energy consumption and the constraint time in data gathering are met. As proposed in [105], an MS is utilized to gather sensed data in the network. During data gathering, the residual energy information of the CHs is also collected by the MS for scheduling its next movement strategy. However, the MS always moves towards energy-rich areas, thereby causing huge data delivery latency from those areas where more events occur.

Another interesting idea for the data collecting schedule is proposed in [146]. In that study, Tashtarian et al. proposed an energy-efficient data collection algorithm that collects data by using the mathematical model of Mixed Integer Linear Programming (MILP). However,
if the MS has not enough time to visit all CHs in the network, the transmission ranges of all CHs must increase to create two or more “range overlapped” areas. Therefore, the MS can save its traveling time by moving to these range overlapped areas to collect data. Unfortunately, the energy consumption of the CHs increases exponentially as their radio transmission ranges grow. The long radio transmission range may disrupt communications and increase the data transmission delay. Moreover, it is not reasonable when all CH transmission ranges increase at the same level while they have different residual energy. Therefore, in my approach, the CH transmission ranges are increased only after maximizing the speed of the MS, and the transmission range of each CH will be increased on the basis of its residual energy.

Many authors utilized the Traveling Salesman Problem (TSP) to find the optimal trajectory of the MS [4, 43]. In these works, a node may be visited more than once before all other nodes are visited. However, in my model, I keep the original TSP technique to find the optimal trajectory, but the time cost of the MS is the total time for moving and collecting data. In this chapter, I investigate the problem of controlled mobility to find optimal movement strategy for the MS, which can reduce the total communication cost within a predefined reporting strategy for each data transmission round.

The contributions of this chapter are highlighted as follows:

* I performed a detailed review of the literature about CH election and utilizing mobility technology in WSNs, which constitute the main features of this work.

* Unlike previous solutions for CH election, my proposed CHE algorithm tries to find the best candidate node to become a CH for each geographical region, which has both high residual energy and low communication cost. This enables us to investigate the effect of residual energy and balanced energy among nodes on network lifetime improvement. The average improvement in network lifetime of CHE algorithm versus the Impro-LEACH algorithm and LEACH algorithm is up to 10.64% and 26.2%, respectively.

* In this chapter, I introduce some solutions to find the optimal trajectory of the MS. The Optimal Movement Strategy (OMS) algorithm exhibits better performance than conventional strategies. The simulation results show that my proposed OMS algorithm improves the lifespan of the network compared with random, static, and fixed-trajectory scenarios.

### 3.2 System Model and Assumptions

#### 3.2.1 Basic Assumptions

Before presenting my algorithms, I first specify the general assumptions about the WSN model I used in this study.

- We consider a network with $N$ homogeneous sensor nodes randomly distributed geographically. They have a limited initial energy $E_0$ and are stationary after deployment.

- An MS node can travel freely in the sensing field with unconstrained energy and storage capacity. During the moving time, the MS does not receive any data packets, because the CHs transmit their data only when they can communicate directly
with the MS. I assume that the MS periodically returns to the support center for recharging. Therefore, the energy consumption of the MS operation and movement does not affect the network lifetime.

- The geographic position of the network nodes is known to the MS and used to find the optimal trajectory of the MS.

- In this chapter, I define the network lifetime of a sensor network as the number of rounds until a certain percentage ($\phi = 85\%$) of the network nodes, whose power source is depleted.

- CHs transmit their buffered data to the MS during a specified time interval called the reporting time denoted by $\xi$.

- To evaluate the network lifetime, round ($T_T$) is a cycle in which the MS successfully receives sensed data from all nodes in the network:

$$T_T = \zeta + \sum_{i=1}^{N} \tau_i + \sum_{i=1}^{M} t_i + \sigma(t). \quad (3.2)$$

where $\zeta$ indicates the cluster election time for $M$ areas, $\tau_i$ is the data collection time from sensor node $S_i$ to its corresponding cluster head node $CH_i$, $t_i$ indicates the period of time the MS stops at $CH_i$ until all buffered data can be collected, and $\sigma$ is the total traveling time of the MS spent between the CHs.

- Each sensor node generates a data sample of $m$-bit data packet every round and needs to transmit it to its CH.

### 3.2.2 Network Model

Figure 3.1 briefly presents a network structure with $N = 30$ sensor nodes deployed randomly in the area of interest $A$. For clustering purposes, area $A$ is divided into $M = 6$ equal parts $\{A_1, A_2, ..., A_M\}$. The number of sub-areas ($M$) mainly depends on the size of the monitored area and the node density ($\rho$), which guarantees that there is no need to send more than one MS to a CH for data gathering. Our objective is to find the best candidate nodes for CH election, which minimizes the communication cost and balances the energy consumption among sensor nodes in the network. Additionally, I also find the optimal movement strategy for the MS, which helps the MS to collect sensing data from all CHs with minimum energy expenditure and within the desired time deadline.

### 3.2.3 Energy Model

Similar to [49], I use a simple model for the radio hardware energy dissipation to calculate the energy consumption. Based on whether the distance $d$ between source node and the destination node is shorter or longer than a threshold distance $d_0 = \sqrt{\varepsilon_{fs}/\varepsilon_{mp}}$, a free space or multi-path fading channel model will be used, respectively. $\varepsilon_{fs}$ and $\varepsilon_{mp}$ denote the free space and multi-path transmitter amplifier model, respectively. The amount of energy $E_{TX}$ consumes at the source node, if it transmits an $m$-bit data packet over a distance $d$ to the destination node, can be calculated by

$$E_{TX} = mE_{elec} + E_{amp}(m, d) = mE_{elec} + m\varepsilon d^X \quad (3.3)$$
where \( \varepsilon = \varepsilon_{fs}; \chi = 2 \) if \( d < d_0 \)
\( \varepsilon = \varepsilon_{mp}; \chi = 4 \) otherwise.
\( E_{elec} \) is the electronic energy, which depends on the
digital coding, modulation, filtering, and spreading of the signal. \( E_{amp}(m, d) \) is the energy
needed by the radio amplifier circuit to send \( m \) bits \( d \) meters. To receive an \( m \)-bit data
packet, the destination node also has to expend \( E_{RX} \) amount of energy, calculated as
\[
E_{RX} = mE_{elec}. \tag{3.4}
\]

The load \( L_k(t) \) of node \( S_k \) during round \( t \) is the total power that the node consumes to
receive and transmit data in that round.
\[
L_k(t) = E_{TX}(t) + E_{RX}(t). \tag{3.5}
\]

The lifespan of one sensor node \( T_k \) refers to the time when its residual energy is less than
a threshold (\( \theta \)). Thus, we have
\[
E_0 - \sum_{t=1}^{T_k} L_k(t) \leq \theta, \{k = 1, ..., N\}. \tag{3.6}
\]

It is clear that the higher the load \( L_k(t) \), the shorter the lifetime of node \( S_k \). Hence,
minimizing the network load and improving the load balancing among the WSN nodes
are prerequisites for achieving the maximum network lifetime.
3.3 Proposed Approach

3.3.1 Cluster Heads Election Algorithm

In this subsection, I focus on the CH election, which helps to save sensor node energy power and prolong the network lifetime. To do this, the CHs should be spread evenly [20]. Therefore, in this study, the sensing field was divided into \( M \) equal regions \( A_i \), \( 1 \leq i \leq M \). These regions are called clusters, and the CH election is performed within each geographical region. For simplicity, without loss of generality, I focus on CH election for \( N_k \) sensor nodes randomly deployed in region \( A_k \). To begin the CH election procedure, every node broadcasts a “HELLO” message to the network by a controlled flooding method [97]. The format of the “HELLO” is shown in Figure 3.2. After receiving the “HELLO” message

<table>
<thead>
<tr>
<th>Node ID ((S_k))</th>
<th>Location ((x_k, y_k))</th>
<th>Total length ((l_k))</th>
<th>Residual energy ((E_k))</th>
</tr>
</thead>
</table>

![Figure 3.2: Format of the “HELLO” message broadcast by the sensor nodes](image)

from source node \( S_k \), one sensor node \( S_c \) knows whether it is inside or outside the \( S_k \)'s region by comparing its location with \( S_k \)'s position. If it is outside the region, it will simply drop the message. If it is inside, the prior values \( PV_k, PV_c \) will be compared with each other (these parameters described later in (3.8)). If \( PV_c \leq PV_k \), \( S_c \) will not broadcast its “HELLO” message and only re-broadcast \( S_k \)'s “HELLO” message. Otherwise, \( S_k \)'s “HELLO” message will be dropped and \( S_c \) will broadcast its own “HELLO” message to the region. As a result, one sensor node knows the ID and location of its CH, whose prior value is the highest in the region. After becoming a CH node, it will announce itself as a CH by broadcasting a CH announcement message (CHAM) to the network. This CHAM contains the CH node’s ID and location. After receiving these CHAMs, one normal sensor node will choose the closest CH \((S_k)\) as its CH node. This enables some sensor nodes, which are in other regions but closer to the CH \( S_k \) than the CH in their regions, to join \( S_k \)'s cluster in order to reduce their energy consumption. The shortest path routing from each cluster node to its CH will be found for data transmission purposes. In order to gain efficiency in using the CH election method, I state the following definition.

**Definition 1.** Assuming that \( N_i \) sensor nodes are deployed randomly over a sub-region of interest \( A_i \). \( E_k \) indicates the residual energy of node \( S_k \). \( l_k \) is the total length of the shortest path routing from node \( S_k \) to all \((N_i - 1)\) nodes in \( A_i \). The energy expenditure of one cluster can be minimal if and only if the candidate node \( S_k \) has

\[
\begin{align*}
\{ l_k &= \min \{ l_1, ..., l_{N_i} \} \\
E_k &= \max \{ E_1, ..., E_{N_i} \} \}
\end{align*}
\tag{3.7}
\]

In practical applications, it is not easy to find one candidate node that satisfies (3.7) in a large area with a large number of CNs. Therefore, in this study, one candidate node will be a CH if it has a maximum priority value, given as

\[
P V_k = \max \left\{ \alpha \frac{1}{l_k} + \beta \frac{E_k(t)}{E_0} \right\}, \quad k = 1, ..., N_i \tag{3.8}
\]
where \( \alpha, \beta \) are positive real numbers such that \( \alpha + \beta = 1 \). These are the weighting factors, which are found heuristically in the course of the optimization. Note that the relative importance of the objectives depends on these heuristic constants \( \alpha, \beta \). In my numerical tests, I ran the CHE algorithm 1,000 times for each set of values \((\alpha, \beta)\). The most profitable values of \((\alpha, \beta)\), which are the best balance between \( l_k \) and \( E_k(t) \), were chosen for the subsequent simulations.

The CH election procedures for each area can be described in more detail by following the steps of CHE algorithm. In Figure 3.3, I illustrate an example CH election process for

### Algorithm 3: The CHE algorithm

1. **Step 1.** Sort the sensor nodes base on the residual energy \( E_i(t), i = 1, \ldots, N_i \).

2. **Step 2.** Compute the shortest path from a single source node (the candidate cluster-head node \( S_k \)) to \((N_i - 1)\) normal sensor nodes: \( R_{k\lambda}^i \), where
   \[
   \begin{align*}
   &k = \{1, \ldots, N_i\} \\
   &\lambda = \{1, \ldots, N_i - 1\} \\
   &\lambda \neq k
   \end{align*}
   \]

3. **Step 3.** Calculate the total length of the shortest path from candidate node \( S_k \) to \((N_i - 1)\) sensor nodes in the sub-region of interest:
   \[
   l_k = \sum_{x=1}^{N_i-1} R_{k\lambda_x}^i, k = \{1, \ldots, L\}, \lambda \neq k
   \] (3.9)

4. **Step 4.** Calculate the priority value \( PV_k \) for each node using Equation (3.10)
   \[
   PV_k = \left\{ \frac{1}{l_k} + \frac{\beta E_k(t)}{E_0} \right\}
   \] (3.10)

   Repeat steps 1-4 for priority value of all cluster nodes in the sub-area \( A_i \) is calculated.

5. **Step 5.** A normal node \( S_k \) will win the competition and becomes a CH node for the sub-area \( A_i \) with the maximum priority value.
   \[
   PV_k : \max\{PV_1, PV_2, \ldots, PV_{N_i}\}.
   \] (3.11)

6. **Step 6.** Repeat steps 1-5 for cluster-head election for all sub-areas in the network.

sub-area \( A_1 \). Figure 3.3 depicts the sensor deployment of \( N_1 = 6 \) sensor nodes in the area \( A_1 \) with the percentage residual energy. In this example, I set \((\alpha = 0.2, \beta = 0.8)\). The priority values of all sensor nodes were calculated, and are given in Table 3.1.

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Node location</th>
<th>( E_i(t)(%) )</th>
<th>( l_i )</th>
<th>Priori value</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>( (16,10) )</td>
<td>90</td>
<td>35.2</td>
<td>0.726</td>
</tr>
<tr>
<td>S2</td>
<td>( (25,7) )</td>
<td>75</td>
<td>33.8</td>
<td>0.606</td>
</tr>
<tr>
<td>S3</td>
<td>( (18,14) )</td>
<td>80</td>
<td>36.1</td>
<td>0.646</td>
</tr>
<tr>
<td>S4</td>
<td>( (26,12) )</td>
<td>60</td>
<td>35.6</td>
<td>0.486</td>
</tr>
<tr>
<td>S5</td>
<td>( (31,7) )</td>
<td>75</td>
<td>40.0</td>
<td>0.605</td>
</tr>
<tr>
<td>S6</td>
<td>( (29,16) )</td>
<td>70</td>
<td>40.7</td>
<td>0.565</td>
</tr>
</tbody>
</table>

34
example, with the highest priority value, node $S_1$ is chosen as the CH, and the shortest path routing of this cluster is described in Figure 3.3.

By analyzing the simulation results, we can state several key features of my proposed CHE algorithm: Firstly, uniform distribution of CHs in each geographic region schemes achieve more energy efficiency and achieve more balanced load distribution than other schemes. Secondly, the number of CHs increases with the size of the monitored area. Therefore, it reduces the packet drop rate and prevents the black hole problem. The last and the most important feature is the election of the CH depend not only on its residual energy but also on the communication cost of the cluster.

### 3.3.2 Optimizing the Trajectory of the Mobile Sink

In this section, I present the MS trajectory optimization (MSTO) algorithm, whose basic idea is to investigate the path with the smallest time spent on traveling and gathering data. To facilitate the MSTO algorithm, I first introduce the following theorem, which helps to construct the infrastructure of the MSTO algorithm.
**Theorem 1.** Let $K_{\text{min}}$ denote the number of MSs used in data collection; $\xi_0$ is a threshold of reporting time, and $R$ indicates the data transmission rate of the MS. It will be an efficient data gathering scheme without loss of sensed data if and only if

$$\frac{1}{K_{\text{min}}} \left( \frac{\mathcal{R}(t)}{v(t)} + \frac{mN}{R} \right) \leq \xi_0$$

(3.12)

Where $\mathcal{R}(t)$ is the total shortest path length of M CHs in the network and $v(t)$ is the velocity of the MS at the current round $t$.

**Proof.** In order to collect data effectively, all monitored data should be collected in time to avoid the buffer overflow problem occurring at CHs. Thus, the reporting time at every cycle time $\xi(t)$ must remain static at the desired value $\xi_0$. Its value at instant round $t$ can be calculated by

$$\xi(t) = \sum_{i=1}^{M} t_i + \sigma(t)$$

(3.13)

If the cluster of $CH_i$ has $V_i$ CNs and $R$ indicates the data transmission rate between a CH node and the MS, the time spent by the MS for data collection at $CH_i$ will be

$$t_i = \frac{mV_i}{R}$$

(3.14)

where $m$ denotes the data packet size generated by each sensor node in one round. Total traveling time $\sigma(t)$ of the MS at the current round $t$ can be computed as

$$\sigma(t) = \frac{\mathcal{R}(t)}{v(t)}$$

(3.15)

By inserting (3.15) and (3.14) into (3.13), the reporting time can be rewritten as

$$\xi(t) = \frac{mN}{R} + \frac{\mathcal{R}(t)}{v(t)}.$$

(3.16)

Thus, the total time spent by $K_{\text{min}}$ MSs is given as

$$\xi(t) = \frac{1}{K_{\text{min}}} \left( \frac{\mathcal{R}(t)}{v(t)} + \frac{mN}{R} \right) \leq \xi_0$$

(3.17)

If $\xi_0$ is high enough, a single MS with constant velocity can harvest all the captured data from $M$ CHs in order to reduce the communication cost. Unfortunately, in practical applications, this is not enough time for data collection. In this situation, we have two scenarios to eliminate this problem: (i) Scenario 1. change the speed of a single MS; (ii) Scenario 2. utilize more than one MS to collect data.

The problem here is how to reduce the total length of the MS’s trajectory in order to minimize the execution time and the energy dissipation for data collection [69]. With these requirements, the optimal trajectory of the MS can be formulated as follows.

**Objective:**

$$\max \left\{ \frac{1}{K_{\text{min}}} \left( \frac{\mathcal{R}(t)}{v(t)} + \frac{mN}{R} \right) \right\} \leq \xi_0$$

(3.18)
Subject to:

\[
C_1 : 1 \leq K_{\text{min}} \leq M \\
C_2 : 0 < t_i \leq \xi_0, \text{for all } i \text{ with } 1 \leq i \leq M \\
C_3 : \mathcal{R}(t) = \{CH_1, ..., CH_M\}. 
\]  

Constraint (C1) ensures that there is no need to send more than one MS to one CH at the same time for data gathering. Constraint (C2) restricts the data collection time from one CH to the MS. Finally, (C3) is the shortest path routing between \( M \) CHs in the current round \( t \).

Implementing the objective given in (3.18), the velocity of the MS or the number of MSs will be calculated depending on the chosen scenario.

\* Scenario 1. In this scenario, a single MS \( (K_{\text{min}} = 1) \) moves along the shortest path of all CHs to collect sensed data. Its velocity \( v(t) \) will now vary according to the change of the shortest path length \( \mathcal{R}(t) \):

\[
v(t) \geq \frac{\mathcal{R}(t)R}{\xi_0R - mN} 
\]  

However, if \( \mathcal{R}(t) \) is too long and \( v(t) \) may be greater than a predefined speed \( v_{\text{max}} \), the speed of the MS will be kept at \( v_{\text{max}} \) in these cases. The data transmission range of some CHs can then be changed in order to reduce the total length of the shortest path of CHs. Unfortunately, the larger a CH’s radio transmission range, the longer the process of data packets toward their final destinations. Therefore, the energy dissipation by each CH with the radio transmission range grows exponentially as given in [29]. The problem here is how to find the transmission range of each CH.

To answer this question, I present a solution that can find the best data collection places for the MS after changing the transmission range of each CH. This provides better results than the approach proposed in [146], which may lead to an imbalanced energy dissipation of the sensor nodes in a WSN when the transmission range of all CHs are increased equally.

For simplicity, I consider a sensing field with six CHs as depicted in Figure 3.4. As shown in Figure 3.4a, if the total length of the MS’s trajectory \( (\mathcal{R}(t) = O_1O_2 + O_2O_3 + O_3O_6 + O_6O_5 + O_5O_4 + O_4O_1) \), where \( O_i \) denotes the best place for the MS to gather data from \( CH_i \), is less than the longest length \( (\mathcal{R}_{\text{max}} = \xi_0v_{\text{max}}) \), the MS needs only to change its speed in order to collect the data in time. Otherwise, the communication range of each CH will be changed in order to decrease the MS’s trajectory. In this case, two parameters include the new transmission range of each CH and the new optimal trajectory of the MS are recomputed.

(i) Evaluating CH transmission range

Let us assume that \( E_i(t) \) is the residual energy of \( CH_i \) with \( N_i \) CNs at the current round \( t \), and \( E_{DA} \) is the energy consumption for data aggregation. The maximum distance that \( CH_i \) can transmit \( N_im \) bits to the MS is

\[
d_{\text{max}}^i = \sqrt{\frac{E_i(t) - [(2N_i - 1) E_{elec} + N_imE_{DA}]}{N_i m\varepsilon}}. 
\]  

The transmission range \( (tr_i) \) of the \( CH_i \) will be increased by

\[
tr_i = \lambda d_{\text{max}}^i 
\]  

\[37\]
Sink movement strategy for $\mathcal{R}(t) \leq \mathcal{R}_{\text{max}}$

Sink movement strategy for $\mathcal{R}(t) > \mathcal{R}_{\text{max}}$

Figure 3.4: Sink movement strategy for Scenario 1.

where $\lambda$ is a coefficient that increases from 0 to 1. According to (3.22), the increase in radio transmission range of each CH depends on its residual energy. Consequently, it guarantees the energy balance among sensor nodes in the network.

(ii) Evaluating the best data collection places for the MS

If $\mathcal{R}(t) > \mathcal{R}_{\text{max}}$, the coefficient $\lambda$ will be increased from 0 to 1 in order to increase the CHs transmission ranges until $\mathcal{R}(t) = \mathcal{R}_{\text{max}}$. After each increase step of $\lambda$, the trajectory of the MS will be recomputed, and the best data collection places for the MS now are $H_i, i = 1, ..., 6$. The change in the MS’s trajectory is shown in Figure 3.4b. After changing the transmission ranges of the CHs, the total length of the MS’s trajectory now can be calculated as follows: $\mathcal{R}_s(t) = H_1H_2 + H_2H_3 + H_3H_6 + H_6H_5 + H_5H_4 + H_4H_1$. It is easily proven that $\mathcal{R}_s(t) \leq \mathcal{R}_{\text{max}}$, and the proof is given in Appendix A.2. If $\mathcal{R}_s(t) = \mathcal{R}_{\text{max}}$, the real transmission range of the $CH_i$ will be $r_i = O_iH_i$.

* Scenario 2. In this scenario, the velocity of the MSs is kept constant at a low level, and the number of MSs needed for data gathering is $K_{\text{min}}$:

$$\frac{1}{K_{\text{min}}} \left( \sum_{i=1}^{K_{\text{min}}} \mathcal{R}_i(t) \frac{v_i(t)}{v(t)} + \frac{mN}{R} \right) \leq \xi_0$$

where $\mathcal{R}_i(t)$ is traveled path length of the MS $i$th.

Hence, the main problem is to schedule the movements of $K_{\text{min}}$ MSs, which start and end at a single depot, all CHs are visited within the desired time deadline. I assume that this single depot is the Network Control Center (NCC), which can monitor and control all operational parameters of the MSs. These such operational parameters include $K_{\text{min}}$, velocities $v_i(t)$, and trajectories $\mathcal{R}_i(t)$ of the MSs in order
to avoid conflicts among them. In this study, the NCC is located at the center of the network field. Similar to the multiple traveling salesman problems (mTSP) [26], each CH must be visited exactly once by a MS in each round. The strategy of each MS is optimized by minimizing the total cost traveled and within the reporting time. For simplicity without loss of generality, the mTSP in this case can be determined on a graph $G = (V, E)$, where $V$ is the set of $M$ CHs and $E$ is the set of edges. Let $C = (t_{ij})$ denotes a cost (execution time for data propagation and the MS’s traveling from node $i$th to node $j$th) matrix associated with $E$. I define the following binary variable:

$$
  x_{ij} = \begin{cases} 
1 & \text{if edge}(i, j) \text{ is used on the tour,} \\
0 & \text{otherwise.}
  \end{cases} \quad (3.24)
$$

Then, a general scheme of the assignment-based directed integer linear programming formulation of the mTSP for one MS ($k$th) can be given as

$$
  \max \left\{ \sum_{i=1}^{M_k-1} \sum_{j=i+1}^{M_k} t_{ij}x_{ij} \right\} \leq \xi_0 \quad (3.25)
$$

$M_k$ is total number of CHs is visited by MS $k$th; subject to:

$$
  \sum_{i=1}^{M_k} x_{ij} = 1, j = 1, ..., M_k \quad (3.26)
$$

$$
  \sum_{i=1}^{M_k-1} \sum_{j=i+1}^{M_k} x_{ij} = M_k \quad (3.27)
$$

$$
  \sum_{i=1}^{K_{min}} M_i = M \quad (3.28)
$$

$$
  x_{ij} \in \{0, 1\}, \forall (i, j) \in E. \quad (3.29)
$$

$$
  0 < t_{ij} \leq \xi_0 \quad (3.30)
$$

Constraint (3.26) forces every CH to be visited once. Constraints (3.27) and (3.28) imply that no more than $K_{min}$ MSs are utilized for data collection. Constraint (3.27) ensures that it is not necessary to use more than one MS to collect data from one CH. Figure 3.5 illustrates the MSs’ trajectories in three sample rounds ($t_1$, $t_2$, and $t_3$). In round $t_1$ (see Figure 3.5a), the distances between MSs are rather far, therefore, four MSs are utilized to collect data from all CHs in order to finish this work within $\xi_0$. In rounds $t_2$ and $t_3$, as shown in Figures 3.5b and 3.5c, these distances are closer, therefore, the number of utilized MSs is two or three MSs, respectively. We now present an outline of the basic heuristic algorithm to find the optimal trajectory of the MS. The procedures of the MSTO algorithm are explained in details in Algorithm 4.
3.3.3 Optimal Movement Strategy for Mobile Sink

In this section, I provide details of the optimal movement strategy (OMS) for a MS. The pseudo code of the OMS is presented in algorithm 5.

3.4 Performance Evaluation and Discussion

In this section, I provide simulation results of my algorithm performed in a MATLAB environment.
Algorithm 4: The MSTO algorithm

1. Starting tour \((T_k)\) for the \(MS_k\) at the Network Control Center by setting \(T_{ex}^k = 0\) \((T_{ex}^k\) is the best improvement so far for total execution time of \(MS_k\)).

2. Pick the closet cluster-head node \((CH_p)\) to the NCC, which is not in visited list of the MSs, to start.

3. Sort all edges.

4. Estimate the \(T_{ex}^k\) after choosing the closest cluster-head node \((CH_c)\) for next stop of the tour. \(T_{ex}^k = T_{ex}^k + t_p + t_c + \frac{d_{p,c}}{v}\) where \(d_{p,c}\) is the distance between \(CH_p\) and \(CH_c\), \(v\) is the velocity of the MS, \(t_p, t_c\) are the total spent time of a MS to collect data from \(CH_p\) and \(CH_c\), respectively.

   * If \(T_{ex}^k > \xi_0\), reject the \(CH_c\) and continue estimating the \(T_{ex}^k\) after choosing the further CH. If all choices \(((M - 1)\) choices) are exhausted without profit, return Step 1 with better starting point.

   * Otherwise, accept the \(CH_c\), add it to the tour and marks this CH as a visited node of the MS.

5. Repeat Steps 3 and 4 in order of increasing length of the tour, as long as they satisfy the execution time criterion \(T_{ex}^k \leq \xi_0\).

6. Check if all \(M\) cluster-head node is visited? If no increase the number of MS \(k = k + 1\) and return the Step 1. And if yes, terminate the algorithm.

3.4.1 Simulation Environment

In my simulations, the parameter settings are summarized in Table 3.2. One normal node is called a “dead node” if its residual energy is less than \(\theta = 0.1(mJ)\).

3.4.2 Numerical Results and Discussion

(i) Cluster Head Election-Performance Analysis and Experimental Comparison

For using the CHE algorithm efficiently, I conducted some experimental studies in a static network. \(N = 150\) sensor nodes were deployed in an area of \(300 \times 300 (m^2)\). A single static BS was located at \(B_1(150, 350)\). Herein, for comparison between methods, after collecting data from the sensor nodes, each CH node transferred its data to the BS directly.

Running a set of simulations with different input values to find the best profit weighting factors \((\alpha, \beta)\). The simulation results, are given in Table 3.3, indicated that the set \((\alpha = 0.6, \beta = 0.4)\) is the best solution with the highest network lifetime. Therefore, I used these values of \((\alpha, \beta)\) in the subsequent simulations.

The performance of CHE algorithm was evaluated and compared with the method proposed in [50] and [165], when changing the network size. It is proved that LEACH could lengthen the network lifetime when compared with direct transmission and minimum-transmission-energy routing. However, as can be seen in Figure 3.6, the network lifetime of the LEACH algorithm is limited to \(t = 1261\) (rounds), whereas for the CHE algorithm, this number is \(t = 1631\) (rounds). Therefore, my proposed CHE algorithm improves the network lifetime up to 29% compared with the LEACH algorithm. Although the Impro-LEACH proposed in [165] has better results with
Algorithm 5: The Optimal Movement Strategy for MS

1 While \( \min_{1 \leq k \leq N} E_k(t) \geq \theta \) do

2 Cluster head election by CHE algorithm.

3 CHs broadcast announcements and one normal node will joint to the cluster with the strongest receive signal strength.

4 According the shortest path routing, normal nodes will transmit or forward monitored data to their CHs. The residual energy level of each sensor node will be calculated by: \( E_k(t) = E_0 - L_k(t), k = 1, \ldots, N \).

5 Compute the minimal spanning tree routing between \( M \) cluster-head nodes in the sensing field.

6 Choose the scenario based on the total shortest path length, and the requirement of reporting time:
   - **Scenario 1**: Set \( K_{\min} = 1 \) and calculate the velocity of the MS by Equation (3.20).
   - **Scenario 2**: Set the velocity of the MSs is the constant value and find the number of the MSs need to be used for data collection by Equation (3.24).

7 Find the optimal trajectory of MS(s) by the MSTO algorithm.

8 After collecting sensed data from all CHs, increase the number of rounds for estimating the network lifetime \( t = t + 1 \); Go back to Step 1.

9 End While

1454 (rounds) and this number is higher than that of the LEACH algorithm, its lifetime is still lower than that of the CHE algorithm. With 177 rounds higher, the network lifetime by the CHE algorithm rises 12\% compared to Impro-LEACH algorithm. However, it is worth noting here that my algorithm improved the network lifetime by balancing the energy consumption among all sensor nodes in the network. For evaluating the performance of the CHE algorithm, Impro-LEACH algorithm, and LEACH algorithm, I conducted experimental studies with different numbers of sensor nodes (varying from 50 to 500), which were deployed randomly in the sensing field of \( 300 \times 300 \) m\(^2\). The network lifetime improvement \( L_i \) in these experimental studies can be computed as given in (3.31)

\[
L_i = \left( 1 - \frac{\text{Lifetime by LEACH or by Impro-LEACH}}{\text{Lifetime by CHE}} \right) \times 100\%.
\]  

The network lifetime comparison of CHE algorithm, Impro-LEACH algorithm, and LEACH algorithm is shown in Figure 3.7. It can be seen that the CHE algorithm has a longer lifespan than the LEACH algorithm and Impro-LEACH algorithm in most of the cases. The improvement achieved in network lifetime by the CHE algorithm is 26.2\% and 10.64\% compared with the LEACH algorithm and Impro-LEACH algorithm, respectively.
### Table 3.2: The settings of simulation parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node deployment</td>
<td>Random &amp; Uniform</td>
</tr>
<tr>
<td># initial energy ($E_0$)</td>
<td>0.1 ($J$)</td>
</tr>
<tr>
<td>$E_{elec}$</td>
<td>50 ($nJ/bit$)</td>
</tr>
<tr>
<td>$\varepsilon_{fs}$</td>
<td>10 ($pJ/bit/m^2$)</td>
</tr>
<tr>
<td>$\varepsilon_{mp}$</td>
<td>0.0013 ($pJ/bit/m^4$)</td>
</tr>
<tr>
<td>Maximum speed ($v_{max}$)</td>
<td>25 ($m/s$)</td>
</tr>
<tr>
<td>Velocity of the MSs in Scenario 2: ($v_2$)</td>
<td>15 ($m/s$)</td>
</tr>
<tr>
<td>Packet length ($m$)</td>
<td>4000 ($bits$)</td>
</tr>
<tr>
<td>Transmission range ($r$)</td>
<td>30 ($m$)</td>
</tr>
<tr>
<td>Data transmission rate ($R$)</td>
<td>250 ($Kb/s$)</td>
</tr>
<tr>
<td>Reporting time ($\xi_0$)</td>
<td>60 ($s$)</td>
</tr>
<tr>
<td>Energy for data aggregation ($E_{DA}$)</td>
<td>5 ($nJ/bit$)</td>
</tr>
</tbody>
</table>

### Table 3.3: Network lifetime (rounds) by CHE algorithm with different values of ($\alpha, \beta$)

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>lifetime</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>lifetime</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1008</td>
<td>0.55</td>
<td>0.45</td>
<td>1528</td>
</tr>
<tr>
<td>0.1</td>
<td>0.9</td>
<td>1167</td>
<td>0.6</td>
<td>0.35</td>
<td>1554</td>
</tr>
<tr>
<td>0.15</td>
<td>0.85</td>
<td>1195</td>
<td>0.65</td>
<td>0.3</td>
<td>1536</td>
</tr>
<tr>
<td>0.2</td>
<td>0.8</td>
<td>1248</td>
<td>0.7</td>
<td>0.3</td>
<td>1568</td>
</tr>
<tr>
<td>0.25</td>
<td>0.75</td>
<td>1281</td>
<td>0.75</td>
<td>0.25</td>
<td>1521</td>
</tr>
<tr>
<td>0.3</td>
<td>0.7</td>
<td>1316</td>
<td>0.8</td>
<td>0.2</td>
<td>1521</td>
</tr>
<tr>
<td>0.35</td>
<td>0.65</td>
<td>1363</td>
<td>0.85</td>
<td>0.15</td>
<td>1430</td>
</tr>
<tr>
<td>0.4</td>
<td>0.6</td>
<td>1405</td>
<td>0.9</td>
<td>0.1</td>
<td>1312</td>
</tr>
<tr>
<td>0.45</td>
<td>0.55</td>
<td>1491</td>
<td>0.95</td>
<td>0.05</td>
<td>1217</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>1512</td>
<td>1</td>
<td>0</td>
<td>831</td>
</tr>
</tbody>
</table>

(ii) **Performance Analysis of the Mobile Sink Trajectory Optimization Algorithm**

In this subsection, I evaluate the performance of two scenarios of the OMS algorithm (I define scenario 1 as OMS1 and scenario 2 as OMS2) by comparing them with two other conventional strategies: (i) static sink, where a stationary sink node is located at the center of the network $B_0(150,150)$; (ii) random moving strategy, where an MS moves randomly in the sensing field; and (iii) MSs moving along the boundary of the network as proposed in [149].

Figure 3.8 depicts the comparison of the network lifetime between these five schemes. It is clear that with increasing node density, the network lifetime of each scheme decreases, because the more sensors, the more data reported to the BS. However, a smaller number of sensor nodes deployed over a larger geographic area leads to a longer distance between sensor nodes and reduces the network lifetime. As shown in Figure 3.8, my proposed schemes (OMS1 and OMS2) achieve longer lifetimes than the three other strategies. Due to the controlled mobility, OMS1 and OMS2 avoid the hot-spot problem completely, which is the main reason for short network lifetime in the stationary scheme and fixed-trajectory attempt. In the random schemes, the MS randomly moves in the sensing field, therefore, it cannot guarantee balanced energy consumption among sensor nodes in the network. The
OMS2 scheme provides better performance, and the network lifetime in this scheme is always higher than that in the OMS1 scheme. However, it is rather expensive to equip more than one MS in the network. Thus, each scenario of the OMS algorithm should be chosen depending on the specific application requirement.

### 3.5 Conclusions

In this chapter, I examined the optimization problem for maximizing network lifetime in WSNs by CH election and finding the optimal trajectory of the MS. My investigation was
Figure 3.8: Network lifetime variation with different node densities

Based on the number and location of CHs in each round to optimize the travel paths of the MSs. The experimental results demonstrate that my proposed CHE algorithm can greatly improve the network lifetime by 10.64% to 26.20% compared with the Improved LEACH algorithm and LEACH algorithm, respectively. Moreover, in optimizing the MS’s trajectory, my proposed OMS algorithm with two scenarios adapts well to the different network sizes, and improves the network lifespan and exhibits better performance than the random, stationary, and fixed-trajectory schemes.
Chapter 4

Quality-of-Service Routing Protocol for wireless sensor networks

In this chapter, I develop a new algorithm to find the optimal path from Source Node to the Base Station that guarantees to preserve energy in WSNs. Optimality is defined in a constrained sense, in which the minimum energy route is sought (to maximize the lifespan of WSNs) under reliability constraint, meaning that each packet must reach the BS with a given probability. Energy efficiency is going to be achieved by selecting nodes for multi-hop packet forwarding under information, which yields the most evenly distributed energy state over the network after the packet has reached the BS. One new routing protocol also provides good performance with any position of the BS in sensing field. Although there are numerous reports of studies with WSN protocols which can increase the network lifetime such as LEACH, PEGASIS, PEDAP, and PEDAP-PA, some of them still fail to provide energy balancing under reliability constraints. In this chapter, I propose a new algorithm under name HQRA (High Quality of service Routing Algorithm), which is able to find near-optimal paths in WSNs by minimizing energy consumption as well as guarantees a given level of network reliability. The numerical results demonstrate that my proposal outperforms other typical algorithms in terms of energy efficiency in data transmission, reliability, and accuracy in routing flows.

4.1 Introduction

Recent technological developments in wireless communication, Micro-electro-mechanical systems enable many applications in WSNs. However, sensor nodes in WSNs are equipped with non-rechargeable batteries with limited battery power. This leads to short lifetime and poses significant technical challenges [54]. One of the major technical challenges is how to save energy while maximizing information throughput, in terms of sending packets to the BS with a given success rate and low energy consumption. Many authors have tried to address energy-efficient routing protocols, such as Directed Diffusion, LEACH, PEGASIS, and PEDAP. Although most of these algorithms increase the network lifetime significantly, there seems to be no algorithm paying enough attention to increasing the reliability of routing protocols. Therefore, in this chapter, I propose HQRA algorithm, which is able to find minimum energy consumption paths in WSNs and guaranteeing a
given level of reliability, as well. Firstly, I briefly describe some popular routing protocols for the sake of comparing them with my proposed HQRA algorithm. One of the well-known WSN protocols is LEACH [50], which is a clustering-based protocol that randomly chooses some sensor nodes as cluster heads, and other sensor nodes forward packets to the nearest cluster head. In this way, the number of sensor nodes, which communicate directly to the BS, will be reduced significantly. Consequently, LEACH can save power and then prolongs the network lifetime. Another well-known protocol called PEGASIS was proposed in [127]. In that study, each sensor node has information about all sensor nodes in the network, therefore, it can send and receive data from neighboring nodes [65]. Huseyin Ozbur Tan, Ibrahim Korpeo et al. [144] described a data routing algorithm named as PEDAP. In that algorithm, all the sensor nodes are connected into a minimum spanning tree. The Base Station can “see” any sensor nodes in the network. After some operation rounds, the dead sensor nodes will be removed and the routing based on current information of the network will be recomputed. Thereby, when comparing with LEACH and PEGASIS, PEDAP algorithm decreases more energy consumption by minimizing the transmission distance. Unfortunately, besides their relatively high efficiency, these algorithms still have many drawbacks. For example, in LEACH, one normal node will become a CH based on a random number ranged from 0 to 1. It will be better if the cluster heads are close to the BS, however, the energy consumption will exponential growth with distance. Therefore, if these CHs are far from the BS this will decrease the network lifetime [71]. While PEDAP focuses on using Prim’s minimum spanning tree algorithm [63], it fails to achieve high bandwidth utilization [112]. Additionally, many authors have been selected the shortest path routing from the source node to the destination node [53,59] in order to decrease the cost transmission but no reliability criteria have been met. Thus, my concern is to find optimal paths with the maximum probability of successful packet reception at the BS and to prolong the longevity of the network at the same time.

The results of the chapter are given in the following structure: The model of HQRA algorithm is explained in Section 4.2. The Novel reliable routing algorithm to maximize the lifespan is described in Section 4.3 and 4.4. A detail performance analysis of HQRA is described in Section 4.5. Finally, I provide some conclusions and outline the future research plan based on this topic in Section 4.6.

4.2 The model

A possible topology of a WSN is depicted in Figure 4.1. We assume that a single-source node transfers sensed data to the BS through some relay sensor nodes, which are placed randomly with 2D uniform distribution. Similar to some proposed studies [80,82,147], I also assume network has the following properties:

- Single BS is locates at a fixed position;
- The batteries of the BS can be recharged by an external source of electric power and the BS is able to communicate in a single-hop with any sensor node in sensing field (even the furthest ones) by radio;
- The number of deployed sensor nodes in the sensing field is $N$ nodes. These nodes are energy constrained and are stationary after deployment.
Some nodes may not have enough power to route its sensing data toward the BS by single-hop routing, therefore, these nodes will reach the BS through some relay nodes. This communication type is multi-hop routing in WSNs.

If necessary, the nodes can organize themselves into a hierarchy where a node at a given level of the hierarchy receive packets from nodes at a lower level of the hierarchy;

The direction of communication is Node - to - BS (the data acquired by the sensor nodes must be collected by the BS);

Let us assume that there is a Wireless Sensor Network perceived as a 2D graph \( G(V, E) \) with \( V \) represents the set of wireless sensor nodes and \( E \) set of edges in the network;

The probability of successful packet between node \( i \) and node \( j \) is determined by the Rayleigh fading model, given as:

\[
P_{ij} = \exp \left( -\frac{\theta \sigma_z^2}{g_i d_{ij}^{-n}} \right) \tag{4.1}
\]

Where: \( \theta (m^{-n}) \) is the sensitivity threshold; \( \sigma_z^2 (W) \) denotes the noise power; \( d_{ij} (m) \) is distance between node \( i \) and neighbor node \( j \); \( g_i (W) \) is a transmission power on sensor node \( i \), and this transmission power can adaptively be changed.
If the energy in sensor nodes is \( g = \{ g_1, ..., g_{|V|} \} \) the energy distributions in the network will be calculated by:

\[
\rho_i = \frac{g_i}{\sum_{l=1}^{N} g_l}; \quad i = 1, ..., |V| 
\]  

(4.2)

\[ i = 1, \cdots, j \]

### 4.3 A novel reliable routing algorithm to maximize the lifespan

In this section, I first characterize the network energy state and then introduce a new routing algorithm.

#### 4.3.1 Characterization of the energy state of the network

The energy state of the network is characterized by introducing an entropy-like quantity defined as follows:

- the energy state of the nodes are denoted by \( G_i, i = 1, \cdots, |V| \);
- the normalized energies are \( \rho_i := \frac{G_i}{\sum_{j=1}^{V} G_j}, i = 1, \cdots, |V| \);
- and the corresponding entropy is

\[
H(g) = \sum_{i=1}^{V} \rho_i \log \left( \frac{1}{\rho_i} \right) 
\]  

(4.3)

It is clear that the larger \( H(g) = \sum_{i=1}^{V} \rho_i \log \left( \frac{1}{\rho_i} \right) \) is, the more evenly distributed energy state over the sensor nodes is. If the lifespan of WSN is defined as the time till the first node goes flat, then more uniform energy state will maximize the lifespan. As a result, when choosing new paths I want to increase \( H(g) \) in order to obtain a more evenly distributed energy state.

#### 4.3.2 Novel routing algorithm

In order to develop a routing algorithm which is energy efficient and reliable at the same token, let us first recall that the reliability of a path (defined as the probability of reaching the BS over the path) is given as

\[
\prod_{i=1}^{M} P_{ij} 
\]  

(4.4)

Where \( M \) is the number of sensor nodes in the optimal path (\( R_{opt} \)).

**Theorem 2.** Assuming that WSN perceived as a 2D graph \( G(V, E) \) with a given transmission vector \( g \) then maximum reliability path can be obtained by performing the Bellman-Ford (BF) algorithm with the link measure \( \frac{\theta g_i^2}{g_i d_{ij}^2} \).
Proof: When searching for the most reliable path one can write
\[ \mathcal{R}_{\text{opt}} : \max_{\mathcal{R}} \prod_{(i,j)} P_{ij} \sim \min_{\mathcal{R}} \sum_{(i,j) \in \mathcal{R}} -\log \frac{1}{P_{ij}} \]

With the Rayleigh fading model was proposed in Equation (4.1), we have:
\[ \mathcal{R}_{\text{opt}} : \min_{\mathcal{R}} \sum_{(i,j) \in \mathcal{R}} -\log \frac{1}{P_{ij}} \sim \min_{\mathcal{R}} \sum_{(i,j) \in \mathcal{R}} -\log \exp \left( -\frac{\theta \sigma^2}{g_i d_{ij}^{-n}} \right) \sim \min_{\mathcal{R}} \sum_{(i,j) \in \mathcal{R}} \frac{\theta \sigma^2}{g_i d_{ij}^{-n}} \quad (4.5) \]

Note that this optimization function is additive, as a result, the optimum can be reached by performing the Bellman-Ford algorithm Q.E.D.

Since the energy in each sensor node is limited, our objective is to choose a path which provides a predefined reliability parameter \((1 - \varepsilon)\) with minimum energy. On the other hand, we would like to achieve this predefined reliability with the smallest possible transmission energies. This cast routing as a constrained optimization problem by searching for a path when the sum of transmission energies are minimal but the given reliability parameter can be achieved.

In order to solve this problem, I propose the following algorithm:

**Algorithm 6: Energy aware reliable routing algorithm**

1. **Assumptions:** The energy state of the WSN is represented by vector \(G\) where components \(G_i, i = 1, ..., |V|\) indicate the available energy on node \(i\). The transmission energies of the nodes are taken form a discrete set \(g_i \in \{\Delta_1, ..., \Delta_L\}, i = 1, ..., |V|\).

2. **Procedure:**

3. **Step 1.** Let each node select the smallest transmission energy \(\Delta_1\).

4. **Step 2.** Calculate the most reliable path by running the BF algorithm with link measure \(\frac{\theta \sigma^2}{g_i d_{ij}^{-n}}\). We have \(\mathcal{R}_{\text{opt}} : \min_{\mathcal{R}} \sum_{(i,j) \in \mathcal{R}} \frac{\theta \sigma^2}{g_i d_{ij}^{-n}}\).

5. **Step 3.** Check condition \(\prod_{i=1}^{M} P_{ij} \geq 1 - \varepsilon\). If it holds the procedure has been finished as a reliable path has been found with minimum transmission energies.

6. **Step 4.** If not then select the transmission energies as \(g_i := G_i - G_{\min}\), where \(G_{\min} : \min_{i=1,..,|V|} G_i\) and go back to step 2.

The optimality of the procedure above lies in the fact that in the first step I use minimum transmission energies and then in the later iteration of the algorithm I balance the remaining battery power on the nodes by always selecting transmission energies which makes the remaining energies uniform. In this way, the lifetime of the network can be maximized. Furthermore, by repeating the steps until the given reliability parameter has been reached, I also solve the constrained optimization.

The steps of this algorithm has been demonstrated by Figure 4.2. Here the node energies \(G_i, i = 1, ..., |V|\) are indicated near the nodes and the reliability parameter \((1 - \varepsilon)\) have been chosen as 0.92.

By performing the BF algorithm with the measure \(\frac{\theta \sigma^2}{g_i d_{ij}^{-n}}\) and using the smallest transmission energies, the optimal path is \(\mathcal{R}_{\text{opt}} (1, 5, 3, \text{BS})\). Unfortunately, the reliability of this
path \( \prod_{S_{opt}} P_{i5}P_{53}P_{3BS} \leq 1 - \varepsilon \). Therefore, I needed to increase the transmission energy and achieved an optimal path as given by Figure 4.3. It should be emphasized that HQRA algorithm works with the smallest transmission energy of the nodes in the path routing. In this way, it saves energy of sensor nodes as possible to prolong the network lifetime (as given in Figure 4.3a). The transmission energy is only increased when the reliability of optimal path is smaller than a respected value \((1 - \varepsilon)\). Therefore, the energy consumption in each node is minimized while the energy balancing is guaranteed among nodes in the network. Figure 4.3b depicts the new energy distribution of the network after the reliability of the path has been reached. It can be seen that sensor nodes 1, 2, and 4 now have similar residual energy level, which helps to improve the network lifetime.

4.4 Complexity analysis

Since the BF algorithm needs \( O(N^3) \) steps and the maximum number of times when it iteratively has to run is the maximum number of energy levels \( \Delta L \), thus the overall complexity in terms of the number of steps the algorithm requires for execution is \( O(\Delta L * N^3) \).

4.5 Numerical results

In this section, I investigate the lifespan of WSN by using the proposed routing algorithm. This investigation involves both the dynamics of energy consumption and the longevity of WSN according to different criteria. To get a deeper insight into the performance I compare the results with the PEDAP algorithm.

4.5.1 Performance analysis and numerical results for HQRA algorithm

The simulation parameters used in the experiments are indicated in Table 4.1. The aim to evaluate the lifespan of WSNs with \( N = 30 \) sensor nodes are deployed randomly in the
(a). Increasing the transmission energy when the reliability of the optimal path has not yet been reached the respected value \((1 - \varepsilon)\)

(b). The new energy distribution when the reliability of the path has been reached the respected value \((1 - \varepsilon)\)

**Figure 4.3:** The optimizing process of the HQRA algorithm

sensing field. The fading parameters are set as \(\theta = 10^{-2}\); noise energy \(\sigma = 0.1\); Propagation parameter \(\alpha = 2\); The smallest energy step to increase the transmission energy in each sensor node \(\Delta g = 10(\mu J)\). In each operation round, one random node will be chosen as a Source Node (SN) to transmit its packet to the BS through some relay nodes. The distance from SN to BS may be too large for direct data transmission, therefore, the source node will transmit its data to the BS through \(M\) relay nodes \((1 \leq M < N)\) by a multi-hop manner. A sensor node is considered to be dead if its energy is smaller than the smallest transmission energy \(\eta\). In this case the threshold of energy I set \(\eta = 100(\mu J)\). The lifetime is defined as the time from the network starting information transmission to the first sensor node died. The graphs in Figure 4.4 and Figure 4.5 illustrated the optimal paths by PEDAP algorithm and HQRA algorithm. The detailed comparison results between PEDAP algorithm and HQRA algorithm are given in Table 4.2. With the same settings of simulation parameters (e.g., the location of source nodes, the smallest transmission energy), HQRA algorithm achieves a higher probability of successful packets than PEDAP algorithm. One of the most important features of these algorithms which can lengthen the
Table 4.1: The setting of simulation parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network size</td>
<td>100m×100m</td>
</tr>
<tr>
<td>Number of sensor nodes ( N )</td>
<td>30</td>
</tr>
<tr>
<td>Node distribution</td>
<td>Uniform distribution</td>
</tr>
<tr>
<td>Threshold Reliability of Networks ( 1 - \varepsilon = 0.92 )</td>
<td></td>
</tr>
<tr>
<td>The sensitivity threshold ( \theta = 10^{-2} )</td>
<td></td>
</tr>
<tr>
<td>Noise energy ( \sigma = 0.1 )</td>
<td></td>
</tr>
<tr>
<td>The smallest energy step ( \Delta g = 10(\mu J) )</td>
<td></td>
</tr>
<tr>
<td>Initial energy in each sensor node ( G_0 = 10000(\mu J) )</td>
<td></td>
</tr>
<tr>
<td>Energy threshold for a dead node ( \eta = 100(\mu J) )</td>
<td></td>
</tr>
<tr>
<td>( E_{elec} )</td>
<td>50 ( (nJ/bit) )</td>
</tr>
<tr>
<td>The transmit amplifier ( E_{amp} )</td>
<td>100 ( (pJ/bit/m^2) )</td>
</tr>
<tr>
<td>Packet size ( k )</td>
<td>5000 ( (bits) )</td>
</tr>
</tbody>
</table>

The optimal path by PEDAP algorithm

Figure 4.4: The optimal path by PEDAP algorithm

network lifetime is the energy balancing among the nodes [161]. As given in Figure 4.6, by the PEDAP algorithm, there are only 2 sensor nodes consuming most of their energy for data transmission before the death of network, while the levels of residual energy in other sensor nodes are still very high. Consequently, the imbalanced energy consumption of nodes leads to waste energy and shorten the network lifetime. In contrast to PEDAP algorithm, my proposed HQRA algorithm achieves better energy balancing among sensor nodes. All the sensor node in HQRA algorithm used up almost all their energies in order to prolong the lifespan. As illustrated in Figure 4.7, we can observe the effectiveness of energy balancing techniques in improving the network lifetime. Obviously, the HQRA
The optimal path by HQRA algorithm with the node 1 is source node

Figure 4.5: The optimal path by HQRA algorithm

Table 4.2: The Probability of successful packets with the smallest transmission energy

<table>
<thead>
<tr>
<th>Source Node</th>
<th>PEDAP Algorithm</th>
<th>HQRA Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Weight</td>
<td>Probability</td>
</tr>
<tr>
<td>1</td>
<td>139.16</td>
<td>0.850</td>
</tr>
<tr>
<td>2</td>
<td>121.68</td>
<td>0.877</td>
</tr>
<tr>
<td>3</td>
<td>51.57</td>
<td>0.940</td>
</tr>
<tr>
<td>27</td>
<td>136.42</td>
<td>0.871</td>
</tr>
<tr>
<td>28</td>
<td>57.44</td>
<td>0.944</td>
</tr>
<tr>
<td>29</td>
<td>82.78</td>
<td>0.906</td>
</tr>
</tbody>
</table>

algorithm achieves high network lifetime, which is more than twice compared with the PEDAP algorithm. Figure 4.8 depicts the comparison of network reliability between two algorithms with different network sizes. In this experiment, I measured the network reliability as the ratio of the number of successful packets received at the BS and the total number of packets sent from source nodes. In this simulation, I regularly increase the number of sensor nodes from $N = 30$ to $N = 50, 75, 100$ nodes in the same network area ($100 \times 100$ m$^2$) and measure the network reliability in each scenario of network size. The probability of successful packet transfers from the source node to its destination node is calculated by Equation (4.1). The optimal paths are selected based on the basis of their probability are higher than $(1 - \varepsilon)$. It is clear that the network reliabilities achieved by
Figure 4.6: The comparison of the energy balancing among nodes in the network.

Figure 4.7: Comparisons of the network lifetime between PEDAP algorithm and HQRA algorithm.

The HQRA algorithm is higher than the ones achieved by PEDAP, in all cases.

In order to evaluate the performance of my proposed algorithm in network lifetime improvement, I continue running the simulation when increasing the reliability threshold from 0.6 to 0.9. Naturally, we find that the higher of the reliability threshold is, the higher of the energy consumption will be consumed, and the lower of the network lifetime is. However, as can be seen in Figure 4.9, HQRA algorithm proves to be superior in terms of having the longest lifetime. Therefore, our algorithm would provide a good trade-off.
$N$ sensor nodes are randomly deployed in $(100 \times 100)$ $m^2$ area.

**Figure 4.8:** The Network reliability with different network sizes between improving network lifetime and guaranteeing a predefined level of network reliability.

**Figure 4.9:** The trend of network lifetime when increasing the reliability threshold
4.6 Conclusions and directions for future research

In this chapter, I have proposed a new high quality of service routing algorithm for WSNs, which can find the optimal path from the source node to destination node with the smallest energy consumption while it still guarantees a predefined level of reliability. Furthermore, the proposed method can run in polynomial complexity with respect to the number of nodes by recursively using the Bellman-Ford algorithm. The new algorithm proved to be far more energy efficient, therefore it can improve the lifetime of WSNs. Our numerical results also demonstrated that in the case of other algorithms, when the first sensor node dies, the network structure does not remain stable and the network energy consumption increases dramatically. As a result, the network quickly ceased to operate because all the nodes became dysfunctional and run out of energy. The new algorithm can solve this problem by balancing the energy consumption among sensor nodes in the network. It could be developed to further enhance real-world applications, which require high reliability and accuracy such as in health applications. For future work, I intend to evaluate the proposed routing protocol via simulations to assess its performance in improving the QoS of health data transmission.
Chapter 5

Prediction-based outlier detection for wireless sensor networks

Outlier detection technique plays an important role in enhancing the reliability of data communication in wireless sensor networks (WSNs). This field has stimulated many researchers which resulted in various proposals on improving the accuracy of outlier detection algorithms. However, most of the existing outlier detection algorithms reveal some drawbacks, e.g., (a) the number of missed outliers are large, (b) the execution time is long, and (c) they may fail to detect outliers online. In this chapter, I propose an effective method which based on the probability of the First Order Error (FOE) to identify outliers on-line hidden in huge sensed datasets. The results demonstrate that my proposal outperforms other typical methods in terms of the accuracy as well as outlier detection capacity in the flexibility of data type.

5.1 Introduction

Wireless sensor networks have become a promising tools which utilized in many real-world applications related to monitoring, data acquisition, and control. They supply a vast amount of online information which collected from the location that sensors are mounted. However, due to constraints on signal processing and capabilities of WSNs, there are some unusual data (called outlier data) may result from sensor malfunction, process disturbances, human-related errors, and/or a sudden change in the state of the environment. The outliers might seriously affect the accuracy of data analysis which causes model misspecification. Therefore, outlier detection is one of the most important preprocessing steps in any data analytical application. There are many authors have proposed some methods to detect the outliers in order to improve the quality of collected data in WSNs.

In the literature, the Hampel Identifier algorithm is the most widely used which provides an efficient outlier identifier. In HI algorithm, the median and the Median Absolute Deviation (MAD) of a moving window with size 2L+1 are calculated for each observation, where L is the number of observations before and after the current observation. Threshold values for outlierness evaluation are calculated by (Median ±θ * MAD), where θ is the generated parameter with the range from 0 to 5. Any observation falling outside the range of such thresholds is identified as an outlier and is replaced by the median value of the data window. However, the HI algorithm reveals its limitations when working with highly autocorrelated data process. More precisely, it may fail to capture outliers due to strong
autocorrelation [87]. Additionally, in HI algorithm, the standard deviation estimates are replaced by the MAD from the median. However, this MAD scale estimator can behave badly with coarsely quantized data [119]. Despite its importance, most of existing outlier detection methods are still mainly designed for cleaning data and do not take into account the real-time outliers identification problem. This may seriously affect the accuracy of real-time decision making. Moreover, to avoid the risks of anomalies data, we have to detect outliers in the streaming data with high accuracy which results in saving the energy consumption as well as the memory usage and shortening the execution time.

In this chapter, my Outlier Detection method is based on the Probability of the Order Error (ODPOE), which has some advantages over existing outlier detection methods given as follows:

* avoiding the limitations of methods which are based on the median in the neighborhood of points [12] or $3 \sigma$ from the mean of data process to detect outliers. Since the presence of outliers tends to inflate the variance estimate [119], these methods may be misleading in outlier detection.

* based on the FOE for detecting outliers, the proposed algorithm can be implemented on the incoming data series with a high identification rate.

* the algorithm can be used for data streams in order to make real-time decisions in WSNs applications.

As a result, my method is computationally simple, and with small memory requirement. Therefore, it may be suitable for the system, which has limited resources such as limited battery power, limited memory, limited communication capacity, as well as, limited computation and processing capacity. My algorithm achieves better performances in terms of accuracy of identification rate than the HI algorithm and other well-known algorithms. The remainder of this chapter is organized as follows. In Section 5.2, I briefly classify the outlier detection methods and summary of some performance indices. In Section 5.3, I first define the problem statements and the background for my network model. Then I describe my algorithm to detect outliers. The performance metrics of my algorithm are analyzed in Section 5.4. Finally, Section 5.5 concludes my work in this chapter and suggests some potential directions.

5.2 Related works

Outlier detection methods have become one of the primary concerns in WSNs. There are many efforts on improving the efficiency of these such methods which can be classified to [13]: (i) parametric (statistical based) methods, and (ii) non-parametric methods. The former [16, 46, 117, 121, 131] assumes a stochastic distribution for observations. It marks the observations as outliers if there is a significant difference between the observations and the model assumptions. Unfortunately, it may fail to identify outliers in high dimensional datasets. Concretely, the complexity and inaccuracy for estimating increase gradually with the multidimensional distributions of data points [2]. The latter consists of the density-based method, the distance-based method, and the clustering-based method [17, 72, 128]. It outperforms the former in the case of using the distances among points to detect outliers. It computes either dimensional distance between points [72, 128], or the densities of local neighborhood [17] to find the outliers in a dataset. Mahalanobis distance-based [164] is applied to label an outlier. In this way, a point is treated as an outlier if its
Mahalanobis distance exceeds a predefined threshold. Similarly, the authors have also used the density-based method to label outliers. However, these methods may be bounded due to deteriorating in the high-dimensional data [76]. A similar technique is proposed in [72]. In that study, the authors defined an observation as an outlier if the distances among such observation and a number of its nearest neighbors exceeds a predefined distance. However, it is difficult to detect outliers in the dataset which has both dense and sparse dimensions [17, 128].

To overcome the aforementioned drawbacks of outlier detection methods, I combine both of the above methods to propose a new outlier detection method in WSNs, which based on the probability of the FOE. The details of this approach will be further described in the next Sections.

For evaluating the effectiveness of outlier detection methods, the following performance indices have been used:

- There are four possible outcomes when detecting outliers [78], they are given in Table 5.1, and are further described in Figure 5.1.
  - Type 1 (Normal points - NP): The inliers which are identified as inliers (by the outlier detection method).
  - Type 2 (False detection points - FP): The inliers which are identified as outliers.
  - Type 3 (Miss detection points - MP): The outliers which are identified as inliers.
  - Type 4 (Correctly detected points - CP): The outliers which are indeed identified as outliers.

<table>
<thead>
<tr>
<th>Actual Inliers</th>
<th>Identified Inliers</th>
<th>Identified Outliers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual Outliers</td>
<td>NP</td>
<td>FP</td>
</tr>
<tr>
<td></td>
<td>MP</td>
<td>CP</td>
</tr>
</tbody>
</table>

Let $OP$ denote the number of predefined outliers in the system ($OP = MP + CP$).

Let $OD$ denote the number of outliers in the data series, which were identified as outliers by a certain algorithm. Obviously, ($OD = FP + CP$).

Let $IR$ denote the identification rate as follows

$$IR = \frac{CP}{Max\{OD; OP\}} \quad (5.1)$$

### 5.3 Outlier detection methods

The main targets of an outlier detection method in WSNs are short running time and high accuracy rate for identifying outliers. In this section, I briefly describe some effective approaches to detect outliers in time-series received from wireless sensor nodes.

#### 5.3.1 Hampel Identifier

The Hampel Identifier proposed in [44] is regarded to be the most important and most widely used algorithm for detecting outliers [87, 119]. This algorithm is known as a non-linear data cleaning filter, which looks for local outliers in time-series or other streaming
The raw data generated by a wireless sensor node

Four possible outcomes of one outlier detection algorithm

**Figure 5.1:** Possible outcomes of a certain outlier detection algorithm

data sequences. The HI algorithm is a variation of the “three-sigma rule” of statistics, by which an observed sample is identified as an outlier if its value lies outside at least $t_0$-times the standard deviation taken from the median of its neighbor values $[87, 119]$. Whenever one data point is detected as an outlier, it will be replaced by its median local value. The details of this algorithm are illustrated in the following steps:

- **Step 1.** Choose the width of the moving window $K$, $0 < K \leq \left\lceil \frac{k-1}{2} \right\rceil$ from $k$ observations;

- **Step 2.** Compute the median value: $\xi_0 = median \left\{ \bar{\xi} [(i - K) : (i + K)] \right\}$;

- **Step 3.** Estimate the standard deviation for each observation:

$$S_0 = H * median \left\{ abs \left\{ \bar{\xi} [(i - K) : (i + K)] - \xi_0 \right\} \right\}$$

where $H = 1.4826$ as given in [23];

- **Step 4.** Identify $(\xi_i)$ as an outlier if $|\bar{\xi}_i - \xi_0| > t_0 * S_0$ where $2 \leq t_0 \leq 5$, and it is replaced by median $\bar{\xi}_i[i] = \xi_0$.

The advantage of the HI algorithm is to detect local outliers in the moving window. Unfortunately, the MAD scale estimator in the HI algorithm is zero if there are more than a half of observations are identical. As the results of that, the performance of the HI algorithm decreases. Furthermore, the authors $[87]$ have claimed that in the case of the input is an independent and identically distributed process, the HI algorithm may fail to detect the outliers due to highly autocorrelated. As a matter of fact, it is hard to find the suitable threshold for each process model.
Overcoming the aforesaid limitations and improving the robustness in outlier detection, I propose a new algorithm, which is based on the probability of error in the data set to detect the outliers. The superior performance of this method in outlier detection is illustrated in the next sections.

5.3.2 The decision theoretic framework to detect outliers

Let $\hat{\xi}_k$ denote the observed time series with respect to $k$. According to [142] I assume that observed sample at time instant $k$ is correlated with the previous $L$ observations according to the Auto-regressive $AR(L)$ model given as follows:

$$\xi_k = \sum_{j=1}^{L} a_{k-j}\xi_{k-j} + v_k,$$

where

$a_1, a_2, ..., a_L$ are the parameters of the model;
$v_k \sim N(0, \sigma)$ is i.i.d.r.v-s (Independent and identically distributed random variables).
$v_k$ represents a noise caused by any effect (measurement inaccuracies, etc.).

At time instant $t = k$ a random variable $\chi_k$ (additive outlier) can also be generated, which is described by the value $(\beta_k)$ and the probability $(\alpha_k)$ of each observed value $\hat{\xi}_k$ is outlier point.

$$\chi_k = \alpha_k\beta_k$$

where:

$$\left\{ \begin{array}{l}
\alpha_k \in \{0, 1\} \\
P(\alpha_k = 1) = p \\
F(x) := P(\beta_k < x) = 1 - e^{-\lambda x} \\
f(x) = \frac{dF(x)}{dx} \\
G^{(l)}(x) = \frac{dG^{(l+1)}(x)}{dx} = f \ast \cdots \ast f(x) \\
g^{(l)}(x) = \frac{dG^{(l)}(x)}{dx} = f \ast \cdots \ast f(x) \end{array} \right.$$

(* denotes the convolution, $\lambda$ is a predefined constant, and $x$ denotes a predefined threshold).

Then the observed process is:

$$\hat{\xi}_k = \xi_k + \chi_k.$$  \hspace{1cm} (5.5)

One can train a predictor in the absence of outlier $\hat{\xi}_k = \xi_k$, prior to installing the whole system:

$$\tilde{\xi}_k = \sum_{j=1}^{L} w_j\xi_{k-j}.$$ \hspace{1cm} (5.6)

Where: $w_{opt} := \min_w \left\{ E(\xi_k - \tilde{\xi}_k)^2 \right\}$

We have:
\[ E\left( \hat{\xi}_k - \tilde{\xi}_k \right)^2 = E\left( \sum_{j=1}^{L} a_j \xi_{k-j} + v_k - \sum_{j=1}^{L} w_j \xi_{k-j} \right)^2 \]

\[ \rightarrow w_{opt} = a \rightarrow \min_w \left\{ E\left( \hat{\xi}_k - \tilde{\xi}_k \right)^2 \right\} = E\left( v_k^2 \right) = \sigma^2. \]

The prediction value, which the ODPOE algorithm used to compare with each observed value

\[ \eta_k = \sum_{j=1}^{L} w_j \hat{\xi}_{k-j} \quad (5.7) \]

where the weights have already been optimized prior to the implementation without outliers (in the following discussion I assume that the training was successful, i.e., \( w_{i, opt} = a_i, i = 1, ..., L \)).

**Detecting outliers:**

As given in [12, 28], in order to implement a computationally simple and real-time outlier detection, we simply use a threshold decision given as follows:

\[ f\left( \hat{\xi}_k \right) = \begin{cases} \text{outlier if } \left| \hat{\xi}_k - \eta_k \right| > \Delta \\ \text{inlier otherwise} \end{cases} \quad (5.8) \]

Where parameter \( \Delta \) can be optimized according to optimizing the Neyman-Pearson hypothesis testing (see [106])

\[ \Delta_{opt} := \min_{\Delta} P\left( \left| \hat{\xi}_k - \eta_k \right| \geq \Delta | \alpha_k = 0 \right) \quad (5.9) \]

under the order error

\[ P\left( \left| \hat{\xi}_k - \eta_k \right| < \Delta | \alpha_k = 1 \right) = \varepsilon. \quad (5.10) \]

The proposed method can be summarized as follows:

**Algorithm 7: ODPOE algorithm for obtaining \( \Delta_{opt} \)**

1. **Step 1.** Preliminary steps: Collect data to establish the parameters of the statistical model \( \sigma, p, F() \).
2. Select the order \( L \) of the autoregressive data process; and collect observable data.
3. **Step 2.** Calculate \( g^{(l)}(x), l = 1, ..., L \) by using convolution.
4. **Step 3.** Calculate the conditional probabilities

\[ P\left( \left| \hat{\xi}_k - \eta_k \right| \geq \Delta | \alpha_k = 0 \right) \quad \text{and} \quad P\left( \left| \hat{\xi}_k - \eta_k \right| < \Delta | \alpha_k = 1 \right) \]

5. **Step 4.** Solve the optimization problem

\[ \Delta_{opt} := \min_{\Delta} P\left( \left| \hat{\xi}_k - \eta_k \right| \geq \Delta | \alpha_k = 0 \right) \]

6. **Step 5.** Detecting and cleaning the outliers by comparing \( \Delta_{opt} \) with the defined rule.
5.4 Application of the outlier detection

To evaluate the performance of the ODPOE algorithm, I perform experiments on both synthetic and real-world datasets. The comparison between the ODPOE algorithm and the HI algorithm has been made based on (i) analyzing the results when using these approaches to detect outliers in a given example data. Applying these algorithms to test (ii) the generated data, and (iii) real-world datasets. All algorithms in this study are implemented by Matlab software version 8.0 (R2012b) on a PC with 6-GB RAM and 2.4 GHz Intel Core i3 processor.

5.4.1 Numerical analysis on the observed data given in [28]

Let us consider the observed data given in [28]. In that study, the authors have identified three outliers (81.5, 79.5, and 78.8) among the observations. This result is used for comparison with the outcome of the HI algorithm and the ODPOE algorithm.

<table>
<thead>
<tr>
<th>Table 5.2: The observed data in [28]</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.6</td>
</tr>
<tr>
<td>18.6</td>
</tr>
</tbody>
</table>

I detect the outliers in the observed data by the HI algorithm with the window width $K = 5$, and $t_0 = 3$. All parameters in the HI algorithm are given in Table 5.3. As we can see, the HI algorithm only detects two outliers i.e., $\hat{\xi}_4$, and $\hat{\xi}_{12}$ in the observed data, while $\hat{\xi}_{10}$ is detected as inlier.

As mentioned before, due to “a symmetric view on dispersion” [123, 130], and highly autocorrelated data process, the HI algorithm may fail to identify the outliers. As consequently, $\hat{\xi}_{10}$ is identified as an inlier. The reason is two outliers (i.e., $\hat{\xi}_{10}, \hat{\xi}_{12}$) occur in a short time. More concretely, the observed data at $\hat{\xi}_{10}$ we have $|\xi_i - \xi_0| = 55.9$ and $t_0 * S_0 = 64.49$. Obviously, $|\xi_i - \xi_0| < t_0 * S_0$. Hence, $\hat{\xi}_{10}$ is identified as an inlier while it is an outlier.

In order to compare the accuracy in detecting outliers between my proposed algorithm and the Hampel Identifier, in this example, I apply the ODPOE algorithm with $L = 3; a_1 = 0.4457; a_2 = 0.1662; a_3 = 0.2266$; and $\varepsilon = 0.85$. Table 5.4 demonstrates the results of my algorithm. After having the absolute error between observed data and the prediction value for each observation (given in Table 5.4a), calculate the probability of the FOE and the responding values of the outlier thresholds (given in Table 5.4b). With $\varepsilon = 0.85$, it can find $\Delta = 35$, and all of outlier points (i.e., $\hat{\xi}_4, \hat{\xi}_{10}$, and $\hat{\xi}_{12}$) in the observed data are detected by the ODPOE algorithm correctly. By this example, it is proven that the ODPOE algorithm achieves higher performance than that of the HI algorithm.

5.4.2 Numerical analysis on synthetic data

I apply the HI algorithm and my ODPOE algorithm to test a generated datasets, in which some outliers are added using the exponential distribution.

More precisely, in the training phase, at each sensor node, I generated random time-series data with the length is $k = 1000$ using an Autoregressive $AR(L = 3)$ as in the following model $\xi_k = \sum_{j=1}^{L} a_j \xi_{k-j} + v_k$. Where the autoregressive coefficients are set as an annual
Table 5.3: Outlier detection by the HI algorithm

<table>
<thead>
<tr>
<th>Data process</th>
<th>$\xi_0$</th>
<th>$\xi_i - \xi_0$</th>
<th>$t_0 \ast S_0$</th>
<th>Outlier</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\xi_1$</td>
<td>22.6</td>
<td>-</td>
<td>-</td>
<td>NA</td>
</tr>
<tr>
<td>$\xi_2$</td>
<td>28.8</td>
<td>-</td>
<td>-</td>
<td>NA</td>
</tr>
<tr>
<td>$\xi_3$</td>
<td>26.8</td>
<td>26.8</td>
<td>0</td>
<td>18.68</td>
</tr>
<tr>
<td>$\xi_4$</td>
<td>81.5</td>
<td>26.8</td>
<td>54.7</td>
<td>34.25</td>
</tr>
<tr>
<td>$\xi_5$</td>
<td>19.1</td>
<td>24.1</td>
<td>5</td>
<td>12.01</td>
</tr>
<tr>
<td>$\xi_6$</td>
<td>15.2</td>
<td>23.6</td>
<td>8.4</td>
<td>14.23</td>
</tr>
<tr>
<td>$\xi_7$</td>
<td>24.1</td>
<td>19.1</td>
<td>5</td>
<td>20.02</td>
</tr>
<tr>
<td>$\xi_8$</td>
<td>23.6</td>
<td>23.6</td>
<td>0</td>
<td>37.36</td>
</tr>
<tr>
<td>$\xi_9$</td>
<td>9.1</td>
<td>23.6</td>
<td>14.5</td>
<td>22.24</td>
</tr>
<tr>
<td>$\xi_{10}$</td>
<td>79.5</td>
<td>23.6</td>
<td>55.9</td>
<td>64.49</td>
</tr>
<tr>
<td>$\xi_{11}$</td>
<td>18.6</td>
<td>23.1</td>
<td>4.5</td>
<td>62.27</td>
</tr>
<tr>
<td>$\xi_{12}$</td>
<td>78.8</td>
<td>23.1</td>
<td>55.7</td>
<td>49.82</td>
</tr>
<tr>
<td>$\xi_{13}$</td>
<td>23.1</td>
<td>20.1</td>
<td>3</td>
<td>13.34</td>
</tr>
<tr>
<td>$\xi_{14}$</td>
<td>11.9</td>
<td>20.3</td>
<td>8.4</td>
<td>12.45</td>
</tr>
<tr>
<td>$\xi_{15}$</td>
<td>20.1</td>
<td>20.1</td>
<td>0</td>
<td>12.45</td>
</tr>
<tr>
<td>$\xi_{16}$</td>
<td>20.3</td>
<td>20.1</td>
<td>0.2</td>
<td>12.45</td>
</tr>
<tr>
<td>$\xi_{17}$</td>
<td>17.3</td>
<td>20.1</td>
<td>2.8</td>
<td>12.45</td>
</tr>
<tr>
<td>$\xi_{18}$</td>
<td>25.8</td>
<td>20.3</td>
<td>5.5</td>
<td>23.13</td>
</tr>
<tr>
<td>$\xi_{19}$</td>
<td>14.1</td>
<td>-</td>
<td>-</td>
<td>NA</td>
</tr>
<tr>
<td>$\xi_{20}$</td>
<td>26.5</td>
<td>-</td>
<td>-</td>
<td>NA</td>
</tr>
</tbody>
</table>

NA - Not available

mean minimum temperature model in [142]: $(a_1, a_2, a_3 = 0.4457; 0.1662; 0.2266)$ and $\nu_k$ is i.i.d subject to $N(0, 1)$, the observed process is $\xi_k = \xi_k + \chi_k$ where $\chi_k$ is an additive outlier.

In the testing phase, I implement outlier detection algorithms to test the dataset. I generate $N$ outliers and randomly add them into the dataset which is distributed exponentially with parameter $\lambda = 1$. Simulation results are depicted in Figure 5.2. Where Figure 5.2a shows the original time-series data and Figure 5.2b depicts the time-series data after adding outliers.

Figure 5.3 depicts the simulation results of the HI algorithm. It is clear that the HI algorithm has many FP and MP points, which are the main reasons for low IR of the algorithm. While the ODPOE algorithm detects most of the outlier correctly, and it is illustrated in Figure 5.4. The comparison of identification rate between the ODPOE algorithm and the HI algorithm is given in Table 5.5. The simulations were run with different numbers of predefined outliers and the outlier percentages. Herein, the portion of outliers is the proportion of the number of predefined outliers over the size of the dataset. The simulation results demonstrate that the identification rate of all algorithms will be decreased corresponding to the increase of the number of predefined outliers or the portion of outliers. However, with higher values of CP points, the ODPOE algorithm has higher identification rates than those of the HI algorithm.

As expected, 78% of outlier points are identified correctly by my algorithm, while the HI algorithm gets 69% as in Table 5.5. These results are depicted in Figure 5.5. Obtaining
### Table 5.4: The process of outlier detection by the ODPOE algorithm

| Data process | \(\eta_i\) | \(|\xi_i - \eta_i|\) | \(\Delta_{i}^{1}\) | \(\Delta_{i}^{2}\) | \(\Delta_{i}^{3}\) |
|--------------|-----------|----------------|-------------|-------------|-------------|
| \(\xi_1\)   | 22.6      | -              | -           | 0.18        | 5           |
| \(\xi_2\)   | 28.8      | -              | -           | 0.29        | 7.5         |
| \(\xi_3\)   | 26.8      | -              | -           | 0.35        | 10          |
| \(\xi_4\)   | 81.5      | 20.93          | 60.57       | 0.47        | 11.5        |
| \(\xi_5\)   | 19.1      | 35.76          | 16.66       | 0.59        | 15          |
| \(\xi_6\)   | 15.2      | 29.82          | 14.62       | 0.65        | 17          |
| \(\xi_7\)   | 24.1      | 42.94          | 18.84       | 0.71        | 20          |
| \(\xi_8\)   | 23.6      | 16.50          | 7.10        | 0.76        | 22          |
| \(\xi_9\)   | 9.1       | 16.13          | 7.03        | 0.82        | 35          |
| \(\xi_{10}\)| 79.5      | 16.73          | 62.77       | 0.88        | 60          |
| \(\xi_{11}\)| 18.6      | 30.05          | 11.45       |             |             |
| \(\xi_{12}\)| 78.8      | 21.48          | 57.32       |             |             |
| \(\xi_{13}\)| 23.1      | 56.38          | 33.28       |             |             |
| \(\xi_{14}\)| 11.9      | 26.62          | 14.72       |             |             |
| \(\xi_{15}\)| 20.1      | 41.66          | 21.56       |             |             |
| \(\xi_{16}\)| 20.3      | 16.83          | 3.47        |             |             |
| \(\xi_{17}\)| 17.3      | 13.24          | 4.06        |             |             |
| \(\xi_{18}\)| 25.8      | 16.25          | 9.55        |             |             |
| \(\xi_{19}\)| 14.1      | 17.77          | 3.67        |             |             |
| \(\xi_{20}\)| 26.5      | 15.19          | 11.31       |             |             |

\[P\left(\left|\xi_i - \eta_i\right| < \Delta|\alpha_k = 1\right)\]

### Table 5.5: Comparison of the identification rate in outlier detection

<table>
<thead>
<tr>
<th>Outlier</th>
<th>HI algorithm</th>
<th>ODPOE algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>N</td>
<td>OD</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>55</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>109</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>113</td>
</tr>
<tr>
<td>10</td>
<td>200</td>
<td>231</td>
</tr>
<tr>
<td>15</td>
<td>150</td>
<td>168</td>
</tr>
<tr>
<td>15</td>
<td>300</td>
<td>350</td>
</tr>
<tr>
<td>20</td>
<td>200</td>
<td>262</td>
</tr>
<tr>
<td>20</td>
<td>400</td>
<td>481</td>
</tr>
<tr>
<td>50</td>
<td>500</td>
<td>587</td>
</tr>
<tr>
<td>50</td>
<td>1000</td>
<td>1231</td>
</tr>
<tr>
<td>75</td>
<td>750</td>
<td>817</td>
</tr>
<tr>
<td>75</td>
<td>1500</td>
<td>1759</td>
</tr>
<tr>
<td>100</td>
<td>1000</td>
<td>1326</td>
</tr>
<tr>
<td>100</td>
<td>2000</td>
<td>2541</td>
</tr>
</tbody>
</table>

Average 600 0.69 0.78
Figure 5.2: Time-series data with $k = 1000; L = 3$, and addition outliers

Figure 5.3: The number of identified outliers by the HI algorithm

high of $IR$ value, the ODPOE algorithm can correctly detect the outliers in time-series data which is useful for real-time decision making.
5.4.3 Numerical analysis on the real dataset

I investigate the effectiveness of my proposed algorithm when applied on the real dataset from Intel Berkeley Research lab between February 28th and April 5th, 2004 [92]. In a period of 31 seconds, each sensor node records temperature, humidity, light and voltage values once. However, I only focus on detecting outliers in dataset generated on March 6th, 2004. The outlier points in Temperature, Humidity, and Light values are detected by the HI algorithm and the ODPOE algorithm are depicted in Figures 5.6, 5.7, 5.8.
details of simulation results are given in Table 5.6. It shows that with a higher number of correctly detected points, the performance of my proposed algorithm is always better than the HI algorithm. In each environmental parameter, there are almost 90% of outliers in dataset are detected by my algorithm while this number of the HI algorithm is only 60%.

Figure 5.6: Outlier detection in temperature values on March 6th, 2004

Figure 5.7: Outlier detection in humidity values on March 6th, 2004

5.4.4 Numerical analysis on detecting the network violations in the 1999 DARPA dataset

In order to show the efficiency of the proposed algorithm, the ODPOE algorithm is used to detect the network violations by detecting outliers in the network data traffic. It is known that network security has remained a major concern of today’s managers, organizations, military, and the government leaders. Although there are several advancements in this field, unfortunately, the number of online crimes has continued to grow over
the years. By the exploitation of vulnerabilities in Web, mobile, or cloud-based applications, attacks may appear as worm, virus, Trojan horse, malicious, or traffic flooding, which result in Distributed Denial of Service (DDoS) or bring down the network services. In these ways, attackers can steal data from personal accounts or the secret information which threatens the national security. Therefore, detecting the initiation of the attacks is really important for preventing and stopping them. In order to improve the network security service, in this scenario, I detect the network violations in the 1999 DARPA Intrusion Detection Evaluation Dataset [85]. Although there are some limitations [98], this dataset is still one of the most widely used in intrusion detection dataset. In that study, Richard Lippmann, et al. tried to provide a realistic background traffic by launching many attack types on the testbed with hundreds of users on thousands of hosts. The dataset was generated synthetically in five weeks with the first three weeks were reserved for training and the last two weeks for testing. However, in this chapter, I only focus on analyzing the inside and outside TCPDump data files on some days of these testing weeks. I used Editcap [30] and Tshark [154] tools to convert the raw DARPA files into the flow data [90]. And then using the ODPOE algorithm to detect the network violations. The simulation results are given in Table 5.7. When comparing with the list of DoS attacks, which labeled on these days of week fourth and fifth in the 1999 DARPA dataset, we can see that the ODPOE algorithm can capture the starting time, and the duration attacking time of an attacker to attack a victim user. With 15 false alarms, the average detecting accuracy of my proposed ODPOE algorithm is 70.0%. Table 5.8 summarizes the algorithm perfor-

![Outlier detection in light values on March 6th, 2004](image)

**Table 5.6:** Outliers in dataset from Intel Berkeley Research lab on March 6th, 2004

<table>
<thead>
<tr>
<th>Methods</th>
<th>NP</th>
<th>FP</th>
<th>MP</th>
<th>CP</th>
<th>Accuracy</th>
<th>The environmental parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>HI</td>
<td>1838</td>
<td>59</td>
<td>61</td>
<td>92</td>
<td>0.601</td>
<td>Temperature</td>
</tr>
<tr>
<td>ODPOE</td>
<td>1917</td>
<td>12</td>
<td>6</td>
<td>115</td>
<td>0.906</td>
<td>Humidity</td>
</tr>
<tr>
<td>HI</td>
<td>1954</td>
<td>28</td>
<td>26</td>
<td>42</td>
<td>0.60</td>
<td></td>
</tr>
<tr>
<td>ODPOE</td>
<td>1989</td>
<td>5</td>
<td>6</td>
<td>50</td>
<td>0.893</td>
<td></td>
</tr>
<tr>
<td>HI</td>
<td>1864</td>
<td>52</td>
<td>42</td>
<td>92</td>
<td>0.639</td>
<td>Light</td>
</tr>
<tr>
<td>ODPOE</td>
<td>1923</td>
<td>4</td>
<td>11</td>
<td>112</td>
<td>0.911</td>
<td></td>
</tr>
</tbody>
</table>
mance when detecting the network violations on some other days of week 4 and 5 in the 1999 DARPA dataset. We can see that with high identification rate (0.72), the ODPOE algorithm will be a great choice for anomaly detection in data traffic.

5.4.5 Numerical analysis on detecting outliers in the monthly series of the Italian Industrial Production Index from 1981 to 1996

I implement the HI algorithm and the ODPOE algorithm to detect outliers in the monthly series of the Italian Industrial Production Index from 1981 to 1996, which is available in bde9915 data frame of the tsoutlier R package [88]. Table 5.9 shows the investigated data series which has been analyzed by many authors [42, 64, 120]. Therefore, it is really useful for the comparison of the HI algorithm and the ODPOE algorithm. In the HI algorithm, I set \( K = 5, t_0 = 3 \). In the ODPOE algorithm, I set \( L = 3; a_1 = 0.35, a_2 = 0.33, a_3 = 0.31, \varepsilon = 0.92 \). The results are depicted in Figure 5.9, and given in Table 5.9. The simulation results illustrated that my proposed ODPOE algorithm detects 16 out of 18 outliers correctly. However, for the HI algorithm, 8 out of 24 detected outliers are FP.

![Figure 5.9: Outlier detection in the Italian industrial production index 1981–1996](image)

5.4.6 Numerical analysis on detecting online attacks

For detecting online the network violations, I setup a small private network with 15 PCs, which are interconnected through a switch and Internet router. Among these PCs, one for the server, one for network analysis, 10 for normal users, and 3 for attackers. The diagram of this network is given in Figure 5.10.
### Table 5.7: The network violations in DARPA dataset

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Duration</th>
<th>Attacker IP</th>
<th>Victim IP</th>
<th>Labeled</th>
<th>ODPOE</th>
</tr>
</thead>
<tbody>
<tr>
<td>03/29/99</td>
<td>16:13:08</td>
<td>0:00:05</td>
<td>172.16.118.70</td>
<td>172.16.112.100</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>03/29/99</td>
<td>21:34:16</td>
<td>0:00:11</td>
<td>6.238.105.108</td>
<td>172.16.112.50</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>03/30/99</td>
<td>14:54:10</td>
<td>0:00:01</td>
<td>172.16.113.50</td>
<td>172.16.113.50</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>03/30/99</td>
<td>15:51:16</td>
<td>0:11:24</td>
<td>194.27.251.21</td>
<td>172.16.114.50</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>03/30/99</td>
<td>17:49:15</td>
<td>0:03:01</td>
<td>208.240.124.83</td>
<td>172.16.114.50</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>03/30/99</td>
<td>21:04:10</td>
<td>0:00:07</td>
<td>209.1.12.46</td>
<td>172.16.112.100</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>03/31/99</td>
<td>10:13:18</td>
<td>0:28:02</td>
<td>172.16.118.60</td>
<td>172.16.113.50</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>03/31/99</td>
<td>11:30:13</td>
<td>0:11:21</td>
<td>172.16.118.20</td>
<td>172.16.113.50</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>03/31/99</td>
<td>14:54:17</td>
<td>0:00:01</td>
<td>194.7.248.153</td>
<td>172.16.112.50</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>03/31/99</td>
<td>15:51:16</td>
<td>0:00:03</td>
<td>1.12.120.6</td>
<td>172.16.112.100</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>03/31/99</td>
<td>18:29:12</td>
<td>0:10:07</td>
<td>194.27.251.21</td>
<td>172.16.114.50</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>04/01/99</td>
<td>08:26:16</td>
<td>0:00:02</td>
<td>206.47.98.151</td>
<td>172.16.114.50</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>04/01/99</td>
<td>08:26:20</td>
<td>0:00:02</td>
<td>172.16.118.60</td>
<td>172.16.114.50</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>04/01/99</td>
<td>11:00:07</td>
<td>0:01:33</td>
<td>172.16.115.234</td>
<td>172.16.112.100</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>04/01/99</td>
<td>16:49:15</td>
<td>0:08:21</td>
<td>172.16.118.20</td>
<td>172.16.112.50</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>04/02/99</td>
<td>11:30:13</td>
<td>0:14:37</td>
<td>206.48.44.50</td>
<td>172.16.114.50</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>04/02/99</td>
<td>11:38:04</td>
<td>0:13:41</td>
<td>192.182.91.233</td>
<td>172.16.112.100</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>04/02/99</td>
<td>13:06:10</td>
<td>0:00:30</td>
<td>166.102.114.43</td>
<td>172.16.113.50</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>04/02/99</td>
<td>13:50:03</td>
<td>0:00:05</td>
<td>194.7.248.153</td>
<td>172.16.112.100</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>04/02/99</td>
<td>14:13:56</td>
<td>0:00:01</td>
<td>172.3.45.1</td>
<td>172.16.112.50</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>04/02/99</td>
<td>18:16:05</td>
<td>0:03:26</td>
<td>10.20.30.40</td>
<td>192.168.1.1</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>04/02/99</td>
<td>20:57:03</td>
<td>0:01:33</td>
<td>172.16.115.234</td>
<td>172.16.112.100</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>04/02/99</td>
<td>10:26:12</td>
<td>0:00:04</td>
<td>152.204.242.193</td>
<td>172.16.114.50</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>04/02/99</td>
<td>15:01:16</td>
<td>0:31:05</td>
<td>172.16.117.52</td>
<td>172.16.113.50</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>04/02/99</td>
<td>15:26:15</td>
<td>0:01:49</td>
<td>194.27.251.21</td>
<td>172.16.114.50</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>04/02/99</td>
<td>17:13:17</td>
<td>0:08:39</td>
<td>172.16.117.52</td>
<td>172.16.114.50</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>04/02/99</td>
<td>11:57:01</td>
<td>0:00:03</td>
<td>194.7.248.153</td>
<td>172.16.112.100</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>04/02/99</td>
<td>15:53:18</td>
<td>0:00:01</td>
<td>199.227.99.125</td>
<td>172.16.114.50</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>04/02/99</td>
<td>19:41:14</td>
<td>0:00:41</td>
<td>209.30.70.14</td>
<td>172.16.112.50</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 5.8: The network violations in fourth and fifth week of the 1999 DARPA dataset

<table>
<thead>
<tr>
<th>Date</th>
<th>Labeled DoS attacks</th>
<th>Detected by ODPOE</th>
<th>IR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FP</td>
<td>MP</td>
<td>CP</td>
</tr>
<tr>
<td>03/29/1999</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>03/30/1999</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>03/31/1999</td>
<td>5</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>04/01/1999</td>
<td>5</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>04/02/1999</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>04/05/1999</td>
<td>14</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>04/06/1999</td>
<td>10</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>04/07/1999</td>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>04/08/1999</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>04/09/1999</td>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.9: Outlier detection in the Italian industrial production index 1981–1996

<table>
<thead>
<tr>
<th>Time</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1981</td>
<td>96.3</td>
<td>90.4</td>
<td>90.4</td>
<td>94.4</td>
<td>95.2</td>
<td><strong>36.6</strong></td>
<td>96.1</td>
<td>95.6</td>
<td>92.8</td>
<td>77.3</td>
</tr>
<tr>
<td>1982</td>
<td>96.9</td>
<td>90.5</td>
<td>88.5</td>
<td>87.9</td>
<td>90.2</td>
<td><strong>36.9</strong></td>
<td>92.3</td>
<td>88</td>
<td>86.8</td>
<td>77.1</td>
</tr>
<tr>
<td>1983</td>
<td>92.6</td>
<td>79.5</td>
<td>87</td>
<td>86.7</td>
<td>84.8</td>
<td><strong>38.4</strong></td>
<td>90.8</td>
<td>87.7</td>
<td>89.5</td>
<td><strong>74.6</strong></td>
</tr>
<tr>
<td>1984</td>
<td>92.6</td>
<td>78.9</td>
<td>93.3</td>
<td>90</td>
<td>88.6</td>
<td><strong>43</strong></td>
<td>89.3</td>
<td><strong>97.5</strong></td>
<td>89.7</td>
<td><strong>73.7</strong></td>
</tr>
<tr>
<td>1985</td>
<td>92.7</td>
<td>84.5</td>
<td>93.9</td>
<td>88.4</td>
<td>93.9</td>
<td><strong>39.1</strong></td>
<td>91.4</td>
<td>96.5</td>
<td>89.7</td>
<td>77</td>
</tr>
<tr>
<td>1986</td>
<td>92.6</td>
<td>93.6</td>
<td>92.4</td>
<td>91.8</td>
<td>99</td>
<td><strong>37.5</strong></td>
<td>97.9</td>
<td>101</td>
<td>91.3</td>
<td>83.3</td>
</tr>
<tr>
<td>1987</td>
<td>102.2</td>
<td>95.3</td>
<td>96</td>
<td>100.5</td>
<td>100.5</td>
<td><strong>39.3</strong></td>
<td>100.3</td>
<td>103</td>
<td>99.1</td>
<td>86.5</td>
</tr>
<tr>
<td>1988</td>
<td>109.5</td>
<td>94.2</td>
<td>104.2</td>
<td>106.1</td>
<td>100.8</td>
<td><strong>46.9</strong></td>
<td>107.2</td>
<td>104.3</td>
<td>106.3</td>
<td>93.3</td>
</tr>
<tr>
<td>1989</td>
<td>109</td>
<td>97.5</td>
<td>107.5</td>
<td>110.3</td>
<td>104.2</td>
<td><strong>50.3</strong></td>
<td>108.1</td>
<td>112.2</td>
<td>109.5</td>
<td>89.8</td>
</tr>
<tr>
<td>1990</td>
<td>112.6</td>
<td>98.1</td>
<td>110.3</td>
<td>105.8</td>
<td>109</td>
<td><strong>51.1</strong></td>
<td>103.6</td>
<td>113.8*</td>
<td>104.6</td>
<td><strong>88.2</strong></td>
</tr>
<tr>
<td>1991</td>
<td>105.9</td>
<td>97.6</td>
<td>108.6</td>
<td>103.1</td>
<td>110.9</td>
<td><strong>46.3</strong></td>
<td>106.9</td>
<td>112.8</td>
<td>105.3</td>
<td>89.1</td>
</tr>
<tr>
<td>1992</td>
<td><strong>111.5</strong>*</td>
<td>102.4</td>
<td>103.1</td>
<td>109.7</td>
<td>111.1</td>
<td><strong>43.4</strong></td>
<td>104.1</td>
<td>107.1</td>
<td>105.4</td>
<td>87.8</td>
</tr>
<tr>
<td>1993</td>
<td>111.3</td>
<td>98.5</td>
<td>102.4</td>
<td>105.7</td>
<td>103.9</td>
<td><strong>45</strong></td>
<td>104.5</td>
<td>101.9</td>
<td>104.1</td>
<td>92.9</td>
</tr>
<tr>
<td>1994</td>
<td>113.3</td>
<td><strong>97.9</strong>*</td>
<td>110.6</td>
<td>112.4</td>
<td>108.8</td>
<td><strong>52.6</strong></td>
<td><strong>112.9</strong>*</td>
<td>109.2</td>
<td>111.7</td>
<td>99.3</td>
</tr>
<tr>
<td>1995</td>
<td>123.6</td>
<td>98.9</td>
<td>117.5</td>
<td>117.7</td>
<td>113.4</td>
<td><strong>58.5</strong></td>
<td>114.1</td>
<td>117.8</td>
<td>115.5</td>
<td>96.6</td>
</tr>
<tr>
<td>1996</td>
<td>115.6</td>
<td>103.5</td>
<td>115.3</td>
<td>110.1</td>
<td>118.1</td>
<td><strong>52</strong></td>
<td>110.7</td>
<td>118.2</td>
<td>108.1</td>
<td>93.6</td>
</tr>
</tbody>
</table>

* Outlier point detected by HI

The server is connected to the switch through a mirror port, which can observe all activity of the switch. The flow data between a source IP address and the destination IP address is captured and stored in the switch cache and then it will be exported to the collector. One computer is connected to the server to analyze the traffic activities (i.e., network flow and network packet) of all PCs in the network. We assume that all the hardware and software for analysis server are safe and strong enough to prevent any virus, worm, Trojan horse, malicious, or active Botnet, which may lead to the deflection results of the simulation. Normal users are asked to use a list of safe applications and websites, where they can download and upload media files (over TCP), or just surfing the Internet (HTTP). By normal behavior, the average of the user’s network flow remain constantly and these PCs are not always active. The attacks are designed to attack the network both in frequency and the traffic load. They may be User Datagram Protocol (UDP) or Internet Control Message Protocol (ICMP) flood, Transmission Control Protocol (TCP) SYN, or Portscan
attack. Most of them force the victimized system to be overloaded and stop servicing new clients. Therefore, my proposed algorithm will detect the traffic anomaly in order to send an early warning to the network operators. The input data for traffic anomaly detection will be captured by Tshark [30], which are the flow traffic from each sampled IP. The timeseries data from each IP includes: # of bytes received/sent, # of packets received/sent. In order to reduce the data volume, the data will be aggregated at 1 minutes interval. And the proposed ODPOE algorithm will be utilized to detect outliers in every 10 minutes data flow records for each User and among Users in the network. The simulation results are given and described in Figures: 5.11, 5.12.

In Figure 5.11, ODPOE algorithm detected outliers in the set of four parameters (i.e., # of bytes received/sent, # of packets received/sent from a user), which present the data flow at User 1. It is clear that there are 2 periods of time (from $t_1 = 96$ to $t_2 = 300$, and from $t_3 = 386$ to $t_4 = 486$) in which the KB received per second by User 1 dramatically increases. It is very likely a DoS attacked the User 1 during these periods of time. High used bandwidth of a user helps to distinguish when an attacker attacks and takes control of a user’s account. This also helps to identify what type of attack has taken in place, the attack functions, and then the network security can guard against the risk of attackers effectively.

In Figures 5.12, the sampled flow data from every user in every 10 minutes also is collected for outlier detection in order to find attackers among users in the network.

In Figure 5.12, there are some attackers, who tried to attack User 1 and User 3. These security threats made the KB received from these Users respectively increased for a short time or a long time during their data transmission. The Bandwidth used by these Users
Figure 5.11: Detection of data-flow anomaly

Figure 5.12: Outlier detection in traffic flow within the network
as recorded in Figure 5.13 are 26.3% and 21.6%, respectively. They are much higher than that of other users in the network. By increasing the data flow rapidly and then overloads the system network connection, the attackers are able to reach their goals by the disruption of network service and then take control of the victim systems. To prevent these risks, my proposed ODPOE algorithm provides an effective solution to detect the anomaly data flow early and then the administrators can choose the best solutions to protect the systems from DoS or DDoS attackers.

![Figure 5.13: The Bandwidth used by Users in the network](image)

### 5.5 Conclusions and directions for future research

I have introduced a new threshold based real-time outlier detection algorithm. The threshold parameter has been optimized according to the Neyman-Pearson lemma. The new method yields a real-time outlier detection because of its algorithmic simplicity. I have performed an excessive numerical analysis, which has proven that the newly introduced method provides better statistics for outlier detection than the typical method (Hampel Identifier method). My method is appropriate for on-line outlier detection in any applications which need accurate analysis and real-time decision-making as well as low energy consumption and small storage requirement.

For future work, I plan to evaluate the performance of my algorithm in some other realistic application scenarios e.g., outlier detection in financial transactions, fault measurement signal detection in on-line health monitoring system. Moreover, I will exploit some smart data capture techniques, which help my algorithms to detect outliers in real-time with high accuracy rate.
Chapter 6

Position Location technique in Non-Line-of-Sight Environments for Wireless Sensor Networks

In this chapter, a novel algorithm is proposed to provide some Range-based localization approaches in wireless sensor networks. Based on the Received Signal Strength (RSS) from neighbor sensor nodes, a single Moving Beacon scans on the path of steepest ascent to locate the positions of sensor nodes. These approaches not only achieve a high degree of the localization accuracy in Non-Line-of-Sight environments but also decrease the execution time considerably. For performance analysis, I compare the average location error and the average execution time versus the speed of the MB. The numerical results demonstrate that my algorithms achieve the best performance if the MB moves under 12(m/s). The numerical results also exhibit a better performance of my algorithms compared to other localization algorithms.

6.1 Introduction

WSN has wide variety applications ranging from agriculture to industrial and military fields. In these applications, fast and reliable localization of every sensor nodes in the network plays a critical role, which is required by (i) coverage, (ii) events detections, and (iii) routing [11,47,135,143,156]. For example, the coverage problem is concerned with the target localization in WSNs [56,99,100]. On the other hand, localization-based routing protocols, which are based on the physical positions of sensor nodes to aim to find the optimal routes with energy efficiency and stability [3,19,79]. Therefore, sensor localization is one of the primary concerns of present-day WSN technology. There are many localization approaches developed for WSNs and they can be divided into two main categories (depending on the way of obtaining the distance information): (i) range-based and (ii) range-free [143]. The range-based approach [7,108,159] uses the distances or the angles between node-to-node to estimate the locations of sensor nodes. In this category, there are some well-known methods to estimate the distances or angles between sender nodes and receiver nodes by using: Time of Arrival [22,160], Time Difference of Arrival [22,150], Angle of Arrival [108,136], Direction of Arrival [67,101], and Received Signal Strength Indicator [135,143,156]. The range-based approach gives high accuracy, but it is expensive and needs a specific hardware. In contrast to the Range-based approach, the Range-free
approach [18, 48, 109] is known as a low energy consumption and cheap approach, which does not need to measure the distances or angles for localization. This approach based on reference points [18], to count the number of hops from unknown position node to the specific anchor points [109], or utilizing some special protocols for node localization. The range-free localization schemes provide an economical approach for determining the location of nodes. However, the Range-free approach is only useful for applications, which have a low localization accuracy requirement. Because the localization error range of this approach is quite high which may fall in the interval 20% - 40% [140].

In this chapter, a new range-based localization algorithm is proposed which uses the RSS to estimate the distance between unknown position nodes and the moving beacon for locating the positions of these nodes. This RSS based location technique affords two important advantages. First, it requires no complex external hardware (there is only a need for a power detector). Second, there is no need for time synchronization between sender nodes and receiver nodes (which is necessary by TOA methods [68]). However, this approach still has several major disadvantages: (i) this method needs at least three reference points to estimate the location of a sensor node; (ii) the power is used at the beacon node may affect the value of RSS at the receiver; (iii) the RSS-based localization techniques may fail in NLoS environments where the signal strength is influenced by obstacles, temperature, and humidity [151].

The smart scan technique of single moving beacon (MB), the one used in this chapter, attempts to overcome these limitations. In the 2D graph, the transmission range of a WSN sensor node is considered as a unit disk (a circular region with unit radius $r$, where $r$ is communication range of a node). The RSS, that is received by the MB from a sensor node, increases and reaches the maximum value at the center of the disk. This center of the disk may be the position of such node. In this chapter, a GPS equipped beacon is used to explore the terrain. Assuming that the MB can determine its geographic position by using a GPS module. The MB periodically broadcasts its current location information to the sensor nodes, which fall inside its transmission area. Based on the RSS measurements, the MB finds the nearest unknown position sensor node and starts locating the position of this such node by two following schemes: (i) scanning all the points on the circle whose center is the current position of the MB and the radius is the estimated distance from the MB to the nearest sensor node (called the expected position circle (EPC)); or (ii) using the steepest ascent search based on the RSS to locate the sensor node.

This chapter also investigates the velocity of the MB to impact on the performance of the proposed algorithms, which guarantees that all sensor nodes will be located with high precision and within a short period of time.

### 6.2 Related works

It is known that there are two types of localization methods: (i) static localization methods, and (ii) mobile localization methods. In the former type, both the sender nodes and receiver nodes in the network are static. These static localization methods are computationally simple. However, they still have some significant drawbacks such as (i) they require the costly additional hardware with a high-density ratio; (ii) the energy consumption of the system will increase dramatically with the increase of the network size (number of sensor nodes and the size of sensing field); (iii) the distance measurement errors may be large and unstable, it can be caused by fading and noise [14]. Fortunately, the latter type, (mobile localization methods [14, 125, 143, 145, 156]) which utilize the moving beacons
for locating the static sensor nodes, can overcome these drawbacks due to the following reasons:

- Since the MB can move anywhere in the area of interest, one can use a small number of the MBs to localize a large number of static sensor nodes. It is impractical if all sensor nodes to be equipped with the costly hardware, but it is possible to support these additional hardware for a small number of the MBs. In [143], Kuo-Feng Ssu et al. proposed a localization method which uses geometry conjecture. A small number of moving anchor points are equipped with GPS and periodically broadcast beacon messages, including their current position information. Sensor nodes may estimate its location by detecting at least three beacon points from these beacon messages. The other similar approach is proposed in [125]. In this study, Priyantha et al. proposed Mobile Assisted Localization (MAL) method which uses a moving mobile to localize the stationary nodes by measuring the pairwise distances. The simulation results show that the position estimation accuracy can improve with a large number of measurements and with the higher number of reference nodes. MAL method also works better with the large mobile coverage areas. Bin Xiao et al. in [156] proposed another method to estimate the position of sensor nodes based on arriving and leaving information of the single moving beacon.

- Due to the small number of MBs, the energy consumption is rather low. Furthermore, in the mobile localization schemes, the sensor nodes can communicate to the MB directly (single hop communication), therefore, they can reduce their energy consumption significantly.

- It is well-known that the RSS would be seriously affected by many physical phenomena such as reflection, absorption, signal fading in the NLoS environments. [129].

For all the reasons above, the best solution for this problem is a single MB with a rechargeable battery is equipped with a GPS module for localization.

### 6.3 The model

In this section, the system model concerning (i) the system environments together with a couple of assumptions regarding studied network; and (ii) the distance estimation method based on the received signal strength is presented.

#### 6.3.1 System Environments and Assumptions

Let us consider a WSN with $N$ unknown position sensor nodes. The nodes are deployed randomly over a region of interest and are denoted by $\{s_1, s_2, ..., s_N\}$. Without loss of generality, this chapter focuses on 2D scenario (the extension to 3D scenario is straightforward). The location coordinates of sensor node $s_i$ are denoted by $s_i(x_i, y_i) \in \mathbb{R}^2$, $i = 1, ..., N$. Assuming that all sensor nodes have the same communication range $r$ and the MB can communicate directly with the sensor node $s_i$ if and only if its location falls within the disk, the center of which is at the node $s_i$ and the disk radius is $r$. Assuming that a single moving beacon $MB(x_0, y_0)$ with the rechargeable power source, is equipped with a GPS device, which can move through the sensing field with the constant velocity $v$ in order to locate unknown position sensor nodes. To make sure all unknown position Nodes in the sensing field are detected, the $MB$ will travel along
A fixed trajectory in sensing field. During moving time, if the MB captures signal from unknown position node, it stops moving on that trajectory for location detection task. The stop-position will be stored in the memory of the MB. After accomplishing location detection task, the MB will go back to that stop-position for continuing its journey. The aim of the chapter is to find the estimated location $s_i = (x_i, y_i), \{i = 1, ..., N\}$ of all sensor nodes with the smallest location error within an acceptable execution time. The location error is defined as

$$le = \sum_{i=1}^{N} \left( \sqrt{(x_i - x)^2 + (y_i - y)^2} \right)$$

(6.1)

A possible arrangement of sensor nodes on the sensing field is depicted in Figure ??a.

### 6.3.2 RSS-Based distance estimation

The Received-Signal-Strength (RSS) can be obtained during the normal signal transmission by a power detector circuit in the receiver devices without demanding additional bandwidth. Thus the distance estimation methods based on the RSS are inexpensive approaches and can simply be implemented in the receiver devices. However, they are unreliable estimations because the strength of RSS may be unpredictable and affected by obstacles in NLoS environments. In the free space propagation model, the power received $P_r (Watt)$ by one antenna under idealized conditions from the other transmitting antenna with a distance $d$ away is expressed by the well-known Friis equation [129], given as

$$\frac{P_r}{P_t} = \frac{G_t G_r \lambda^2}{(4\pi)^2 d^2 L}$$

(6.2)

where $P_r$ is the received power, $P_t$ is the transmitted power, $d$ is distance between transmitter and receiver, $G_t$ and $G_r$ are the antenna gains of the transmitting and receiving antennas respectively, $\lambda = \frac{f_r}{c}$ is the wavelength of the transmitter signal in meters. Here $f_r$ is the carrier frequency of the signal, $c$ is the speed of the light and $L$ is the system loss factor which is independent of propagation environment $L$ ($L \geq 1; L = 1$ indicates that there is no loss in the system hardware).

However, it is not always realistic to use the free-space propagation model which predicts the RSS in LoS environments. In fact, the propagation of signals is affected by many different modes of physical phenomenon (e.g., reflection, diffraction, and scattering [129]) or by some other physical properties [151]. Therefore, there is a nonlinear transformation between the RSS and the location. As a result, many authors in [60,96,157] proposed that RSS and distance are related by the Log-normal shadow model and they are given by the following formula:

$$RSS[dBm] = P_r[dBm] - P_0[dBm] - 10\eta\log_{10} \left( \frac{d}{d_0} \right) + X_\sigma$$

(6.3)

where RSS is the received signal strength in units of decibels relating to 1 milliwatt unit [dBm], $P_r[dBm]$ is the power of received signals of node, $P_0[dBm]$ is a known reference power value at a reference distance $d_0$ from the transmitter, $\eta$ is a path loss index that depends on the propagation environment (i.e., $\eta = 2$ with free space environment, $\eta = 1.6 - 1.8$ in building Line-of-Sight [129]), $X_\sigma$ is zero-mean Gaussian random variable with standard deviation $\sigma$. Following this formula, the transmitter will be at the center of the transmission circle with the decrease of the RSS.
In this chapter, based on the product manual of DIGI XBee PRO modules [1], it assumes that the RSS range received by the MB is \([-100[dBm], -20[dBm]\]). \(RSS = -100[dBm]\) reported for last good packet received when the beacon is out of sensor’s communication area \(d \geq r\) and \(RSS = RSS_{threshold} = -20[dBm]\) when the beacon and sensor node are the same position \(d = 0\). In this chapter, the location of a sensor node \(i\) can be identified if the RSS, which is received by the MB from that sensor node, satisfies (6.4)

\[
|\text{max}\{RSS_i\} - RSS_{\text{threshold}}| \leq \delta
\]  

(6.4)

where \(\delta, (\delta \leq 1dBm)\) is threshold parameter.

In ideal conditions, the correlation of received signal strength and separation distance are depicted by Figure 6.1a. If the propagation condition is dictated by NLoS environments, the received signal strength may be different values at the same distance from a transmitter. The changes of RSS in the real condition and in the NLoS environments are depicted in figure 6.1b and figure 6.1c, respectively.

### 6.4 Localization algorithms

I examine several possible scenarios to use the RSS, which the MB receives from a node, to detect the location of that such node. The main focus is on improving the steepest ascent methods to estimate the position of the nodes. In this way, we not only reduce the average execution time but also increase the localization accuracy. In order to obtain the efficiency in using the steepest ascent methods for localization, the following lemma is stated.

**Lemma 1.** Suppose \(f : \mathbb{R}^n \rightarrow \mathbb{R}\) is continuously differentiable on the set \(S = \{RSS \in \mathbb{R}^n | f(RSS(x(k), y(k))) \leq f(RSS(x(0), y(0)))\}\) and \(S\) is a closed and bounded set. Then every point \(A(\bar{x}, \bar{y})\) that is a global maximum of the sequence \(\{x(k), y(k)\}\) satisfies:

\[
\begin{cases}
|RSS(\bar{x}, \bar{y}) - RSS_{\text{threshold}}| \leq \delta \\
\nabla f(RSS(\bar{x}, \bar{y})) = 0
\end{cases}
\]  

(6.5)

where \(f(RSS(x(k), y(k)))\) is the RSS measured by the MB at kth iteration \(K(x(k), y(k))\).

The proof of this lemma can be found in Appendix A.1. Based on this lemma, the MB can detect the position of one sensor node if only if the RSS, which is received by the MB at that position from one sensor node, satisfies (6.5).

### 6.4.1 Localization by Expected Position Circle Scan (EPCS)

The MB can communicate directly to a sensor node at an unknown position in the broadcast field of that sensor node. The position of sensor node may be on the expected position circle (EPC), its radius is the distance from MB to the nearest sensor node, and the center is the current position of MB. The EPCS algorithm is based on the RSS measured by the MB when it scans the EPC to find the location of sensor nodes. The localization scheme illustrated by Figure ??b can be summarized as follows:

1. collecting the RSS values at the current position of the MB;

2. finding the highest RSSI from an unknown position sensor node and estimating the distance between the MB and that sensor node;
(a) The communication range in the ideal condition

(b) The communication range in the real condition

(c) The communication range in the NLoS environments

Figure 6.1: The correlation of received signal strength and separation distance
3. scanning on the EPC;
4. locating the sensor node and listing that sensor node into the list of nodes MB was visited.

The details of EPCS algorithm is described by the following steps:

* Step 1. At the current position, the MB can measure $M$, \( M = 0, ..., N \) signals from $M$ neighbor sensor nodes \( \{S_1, S_2, ..., S_M\} \) if it falls into the communication areas of these sensor nodes. Let \( RSS_i; \{i = 1, ..., M\} \) represent the signal strength of node \( S_i \) measured by the MB. For example, in figure 6.2a, at the current position, the MB can measure two RSS values \( \{RSS_3, RSS_6\} \) from \( S_3 \) and \( S_6 \) \( (RSS_6 > RSS_3). \)

* Step 2. Based on the measured RSS values, the MB estimates the distance \( d_k \) from its current position \( MB(x_0, y_0) \) to the nearest unknown position sensor \( S_k \). (The nearest unknown position sensor is the sensor node that satisfies two conditions: (i) it is an unknown position sensor node (It is not in the visited list of th MB); (ii) the RSS received from this sensor node is bigger than the signal from any other unknown position sensor nodes. Here, in the example, the nearest unknown position sensor node is \( S_6 \).

* Step 3. The MB scans and measures the RSS values \( \{RSS_i\} \) of all the points on the EPC (The blue dashed circle in figure 6.2b). Then the MB moves to the position where has the maximum of the RSS value. Check if \( |\max \{RSS_i\} - RSS_{threshold}| \leq \delta \) and go to step 4, if not go back step 1.

* Step 4. The sensor node position will be located at the point with \( |\max \{RSS_i\} - RSS_{threshold}| \leq \delta \) and the MB inserts that sensor node into the visited list for the next steps.

In real environments, the communication systems will be affected by many environmental factors (e.g., temperature, humidity, obstacles). Some locations have LoS environments and some others have NLoS environments. So the irregularity of RSS in these places will be the main reasons for large errors when estimating the distances between the MB and a sensor node. In these situations, Step 2 and Step 3 may be repeated until the condition \( |\max \{RSS_i\} - RSS_{threshold}| \leq \delta \) is met. Figure 6.3a depicts the procedure of the EPCS algorithm in the NLoS environments. Based on the RSS, the MB estimates the distance \( (d_1) \) between its current position and the sensor node. Then the MB moves and scans on the EPC, whose center is the current position of the MB and radius is \( d_1 \). However, the maximum of RSS value at point B on the EPC is still smaller than a predefined threshold \( |\max \{RSS_i\} - RSS_{threshold}| > \delta \) so EPCS algorithm continues from Step 2. The sensor position will be detected (at point C) when the formula (6.4) is satisfied.

In some cases, there are some obstacles between the MB and the sensor node. This causes a decrease in the strength of the RSS and it may be different at the same distance from the transmitter. These may result in repeating procedure steps 1-3 in the EPCS algorithm. Figure 6.3b describes the trajectory of the MB by the EPCS algorithm in this case.

### 6.4.2 Solution by steepest ascent search (SAS)

The SAS algorithm determines the absolute geographical location of sensor nodes based on the steepest ascent path of the RSS, which is received by the MB from the nearest
Sensor node

The moving beacon

Communication area

(a) Position estimation in WSNs by RSS

(b) Localization in WSNs by EPCS

Figure 6.2: Position estimation with moving beacon in WSN

(a) The EPCS procedure in the real environments

(b) The EPCS procedure with obstacles

Figure 6.3: Position detection by EPCS algorithm
unknown position sensor node $S_i$. The basic idea of this method is very simple and is described in the following steps:

* Step 1. Measure the RSS from all neighbor nodes of the MB, which are within range $r$ of the MB $(x_0, y_0)$: $RSS_i(x_0, y_0), \{i = 0, ..., M\}$.

* Step 2. Find the highest RSS from an unknown position node $S_k(x_k, y_k)$ among $M$ neighbor nodes of the MB.

$$RSS_k = \max_{i=0, ..., M} \{RSS_i(x_0, y_0)\}.$$  

* Step 3. Check if the signal captured by the MB from sensor node $S_k(x_k, y_k)$ satisfies (6.4). The algorithm will be terminated and the position of sensor will be the current position of the MB $S_i(x_{MB}, y_{MB})$ and then the MB will insert that sensor node into the visited list for next steps. If not go to Step 4.

* Step 4. Check all possible directions (n) of the MB and choose the direction with the fastest growth direction of the RSS from source node $S_i$. From (6.3), the RSS function of sensor node $S_i(x_i, y_i)$ is computed by

$$RSS(x(t), y(t)) = Pd_0 - 10\eta \log \left(\sqrt{(x(t) - x_i)^2 + (y(t) - y_i)^2}\right) + X_\sigma \quad (6.6)$$

where $(x(t), y(t))$ is the location of the MB at iteration $t$, $\{t = 1, 2, ...\}$. The coordinates of the next stop point of the MB on the SAS path are calculated as follows.

$$RSS(x(t + 1), y(t + 1)) = RSS(x(t), y(t)) + (\lambda_x(t), \lambda_y(t)) \left(\nabla RSS(x(t), y(t))\right) \quad (6.7)$$

where $(\lambda_x(t), \lambda_y(t))$ is the search step size of the $x$ axis, $y$ axis directions at the $t^{th}$ iteration. $(\lambda_x(t), \lambda_y(t))$ will be chosen with the highest RSS direction among $n$ possible directions of the MB.

Step 4 will be repeated until the condition in (6.4) is satisfied.

Figure 6.4a and 6.4b depict trajectories of the MB by the SAS algorithm in the real environments (no obstacles or LoS environments) and the NLoS environments, respectively. It can be seen from the graphs that the number of iterations on the SAS paths will depend on the convergence of the RSS value received by the MB.

6.4.3 Stochastic gradient ascent (SGA) method for localization in WSNs

One of the biggest disadvantages of the gradient ascent methods is having to check all possible directions (n) to find the next step of the MB’s movement in the search direction. For example, the function to be maximized (the log likelihood with independent observations) has the form:

$$f(RSS) = \sum_{i=1}^{n} f_i(RSS). \quad (6.8)$$

The gradient ascent methods need to check all possible (n) directions of the MB in order to choose the best moving direction by evaluating the following equation:

$$RSS_{t+1} = RSS_t + \lambda \sum_{i=1}^{n} f_i'(RSS_i). \quad (6.9)$$
where $\lambda$ denotes the search step size of the searching method.

As a result, the gradient ascent methods work well in the environments with and without obstacles (LoS environments and NLoS environments). However, It takes extremely long time to find the movement direction of the MB if the number of directions $n$ is huge. In order to speed up the process, in this section, the stochastic gradient ascent method to find the trajectory of the MB when it only randomly chooses $m$ possible directions out of $n$, which are generated randomly subject to a Gaussian distribution ($m \leq n$) is presented. The chosen direction will be the direction with the fastest growth of the RSS from source node among $m$ directions.

$$RSS_{t+1} = RSS_t + \lambda \sum_{i=1}^{m} f_i'(RSS_t)$$  \hspace{1cm} (6.10)

The details of the SGA method is described in the Algorithm 8. The trajectory of the MB by SGA algorithm is depicted in figure 6.6a. It is proven that the algorithm always converges to the global optimum of RSS [35].

### 6.4.4 Local obstacle avoidance for the MB in the NLoS environments

The Stochastic steepest ascent is a simple and widely used method. However, it may have the following drawbacks:

- The number of iterations to find the global maximum becomes large if one hits a local maximum or a plateau [27]. These points are described in the figure 6.5, and they are the main reasons for the slow convergence speed of the steepest ascent methods. In this case, if we increase the step size in order to speed up convergence, the estimated error value will be increased.

To avoid getting stuck on a plateau or in one of the local minimum, we can choose some of the following approaches: (i) backtrack to some earlier positions and try different directions; (ii) make a big jump (long and random step size) to try to get
Algorithm 8: The SGA algorithm

1. **Input:** Parameters of model, $\text{RSS}_i(x, y)$, $\text{RSS}_{\text{threshold}}$; $N_f$ is the number of times to find the MB’s direction for one RSS level.

2. **Output:** $\tilde{S}_i(\tilde{x}_i, \tilde{y}_i)$ the estimated location of sensor node $S_i$.

3. Measure the RSS at the current position of the $MB(x_0, y_0)$.

4. Choose the highest RSSI from an unknown position node on the overlapping sensing areas of $M$ sensor nodes with the current position of $MB(x_0, y_0)$.

   $$RSS_k = \max_{i=0,...,M} \{RSS_i(x_0, y_0)\}$$

5. Assuming the position of sensor node $S_i(x_i, y_i)$ and

   $$\begin{cases} x_{MB} = x_0 \\ y_{MB} = y_0 \end{cases}$$

6. **While** $|RSS_k(x_{MB}, y_{MB}) - RSS_{\text{threshold}}| > \delta$ **do**

7. The current position of the MB will be set

   $$\begin{cases} x(t) = x_{MB} \\ y(t) = y_{MB} \end{cases} ; h = 0$$

8. **While** $h \leq N_f$ **do**

9. Compute the gradient $[\nabla (RSS(x(t))), \nabla (RSS(y(t)))]$ with

   $$\begin{cases} \nabla (RSS(x)) = \frac{(x-x_i)}{[(x-x_i)^2+(y-y_i)^2] \ln(10)} \\ \nabla (RSS(y)) = \frac{(y-y_i)}{[(x-x_i)^2+(y-y_i)^2] \ln(10)} \end{cases} \quad (6.11)$$

   where $S_i(x_i, y_i)$ is the position of the source node.

10. Randomly check $m$ possible directions of the MB ($m$ directions are generated randomly from Gaussian distribution) ($m \leq n$). From $m$ directions, choose the best direction with the highest growth of the RSS. $(\lambda_x(t), \lambda_y(t)) := (\lambda_{x_i}(t), \lambda_{y_j}(t))$.

   Choose the search step size $\lambda(t)$ with

   $$RSS(x(t + 1), y(t + 1)) = \max_{(i,j=1,...,m)} \left\{ RSS(x(t), y(t)) + (\lambda_{x_i}(t), \lambda_{y_j}(t)) (\nabla RSS(x(t), y(t))) \right\} \quad (6.12)$$

11. Take a small step moving of the MB in the direction of the gradient to the next point:

   $$\begin{cases} x(t + 1) = x(t) + \lambda_{x_i}(t) \nabla (RSS(x(t))) \\ y(t + 1) = y(t) + \lambda_{y_j}(t) \nabla (RSS(y(t))) \end{cases}$$

12. **If** $RSS(x(t + 1), y(t + 1)) \geq RSS(x(t), y(t))$ **then**

13. If yes, accept moving and locate the position $MB(x_{MB}, y_{MB})$:

   $$\begin{cases} x_{MB} = x(t + 1) \\ y_{MB} = y(t + 1) \end{cases}$$

14. and if not, increase the number of times to find the MB’s direction for one RSS level.

   $h = h + 1$ and the MB returns to the previous position

   $$\begin{cases} x_{MB} = x(t) \\ y_{MB} = y(t) \end{cases}$$
15 End If
16 End while
17 The MB will return to its previous position at iteration \( t \).
\[
\begin{align*}
x_{MB} &= x(t) \\
y_{MB} &= y(t)
\end{align*}
\]
18 End While
19 Locate the position of sensor \( S_i \) is
\[
\begin{align*}
x_i &= x_{MB} \\
y_i &= y_{MB}
\end{align*}
\]

(a) The plateau points on the MB’s trajectory
(b) The local maximum on the MB’s trajectory
(c) The disadvantage points with the RSS value of the steepest ascent methods

Figure 6.5: The disadvantages of the steepest ascent method
into a new section; (iii) search in several directions at once. In the SAS algorithm and the SGA algorithm, the MB randomly chooses the value of \((\lambda_x(t), \lambda_y(t))\) subject to uniform distribution for each iteration and each different direction in order to make a different jump to try to get to a new section of the search space.

- Furthermore, whenever the MB reaches the plateau or a local maximum (the number of times to find the next position of the MB at one RSS level reaches the threshold \(N_f\)), it returns to the previous position (with a lower level of the RSS received by the MB) to find another direction. One more preferable solution which the MB can use to escape these positions is simulated annealing (SA) method. The SA method to find the global maximum of RSS value, which locates the positions of sensor nodes more accurately in both LoS environments and NLoS environments. The details of this method are described in the next section.

### 6.4.5 Simulated annealing method for approximating the global maximum

In the case of applying the Simulated Annealing (SA) method, the MB chooses a random neighboring position at each step: \(MB(x(t+1), y(t+1))\). This position of the MB is within its communication area, and is generated by Gaussian distribution. From the current position \(MB(x(t), y(t))\) of the MB, it measure the RSS at the \(MB(x(t+1), y(t+1))\). The MB will accept moving to the neighboring position if the RSS at this position \((x(t+1), y(t+1))\) is higher than that at the current position \((x(t), y(t))\). However, if the MB reaches the plateau or local maximum, the RSS at the neighbor positions will be equal to or even less than the signal at the current position of the MB. In these situations, the MB may still move to the next position if the acceptance probability is higher than a random value in the range \([0, 1]\). The acceptance probability is calculated by (6.13)

\[
P = \begin{cases} 
1 & \text{if } \Delta RSS_i < 0 \\
\exp\left(-\frac{\Delta RSS_i}{T}\right) & \text{otherwise}
\end{cases} \tag{6.13}
\]

where \(\Delta RSS_i\) is calculated by (6.14), and \(T\) is the parameter called temperature for annealing, which will decrease at each step of annealing. Thus, the probability of accepting a worse solution also decreases with decreasing temperature.

\[
\Delta RSS_i = RSS_i(x(t+1), y(t+1)) - RSS_i(x(t), y(t)) \tag{6.14}
\]

The details of the SA method are depicted by the following flowchart in figure 6.7. The trajectory of the MB dictated by SA is described in Figure 6.6b.

### 6.5 Performance evaluation

The proposed localization algorithms in MATLAB software are implemented and the parameters using in simulations are summarized in Table 6.1.

#### 6.5.1 Simulation design

Assuming that in each Monte Carlo (MC) running time, \(N\) unknown position sensor nodes are randomly deployed over a region of interest \(A(500 \times 500m^2)\). The single moving beacon
Figure 6.6: The trajectory of the MB by SGA algorithm and SA algorithm
can move anywhere in area $A$ and it can communicate directly with any sensor nodes within the transmission range distance. In my simulations, I set the path loss exponent $\eta = 3$, the reference power value $P_0 = 40(dBm)$ at a reference distance $d_0 = 1(m)$ and the transmission range of all sensor nodes is assumed to be $r = 15(m)$. $X_\sigma$ is Gaussian distribution whose mean is 0, and variance $\sigma^2_{dBm} = 3$ (dBm). In order to verify the stability
of the algorithms, each algorithm will be simulated $M_C = 1000$ times and the results will be averaged. For evaluating the EPCS algorithm, SAS algorithm, SGA algorithm and SA algorithm, some following performance indices have been used:

* As given in Equation (6.1), the location error is the distance between the estimated coordinates $\tilde{s}_j (\tilde{x}_j, \tilde{y}_j)$ and real coordinates $s_i (x_i, y_i)$ of sensor node $s_i$. So the Average Location Error (ALE) of all sensor nodes can be calculated as

$$ALE = \frac{\sum_{i=1}^{MC} \sum_{j=1}^{N} \left[ (\tilde{x}_{ij} - x_{ij})^2 + (\tilde{y}_{ij} - y_{ij})^2 \right]}{NM_C},$$

(6.15)

where $\tilde{s}_j (\tilde{x}_{ij}, \tilde{y}_{ij})$ denotes the estimate of the true location $s_j (x_{ij}, y_{ij})$ of sensor node $s_j$ in the $i$th Monte Carlo run.

* The Average Execution Time (AET) of the localization is calculated as

$$AET = \frac{\sum_{i=1}^{N} \text{execution\_time}_i}{N}.$$  

(6.16)

The execution time of localization is reduced by increasing the velocity of the MB. In this way, more beacon messages are received during the same time period. It is worth noting that the speedup of the MB is also the cause of increasing the localization error [84,143] due to the higher probability of mis-selection points with the maximum RSS value. In this study, the speed of the MB will vary from 1(m/s) to 50(m/s).

### 6.5.2 Simulation results

To evaluate the efficiency of the proposed algorithms, I focus on comparing two performance indices (the ALE and the AET) of the SA algorithm, SGA algorithm, SAS algorithm and EPCS algorithm.

Figure 6.8 depicts the results of the EPCS algorithm, SAS algorithm, SGA algorithm and SA algorithm with the velocity of the MB being 10(m/s). It is clear that by choosing the best direction of the MB at each step from all possible directions ($n$), the SAS algorithm gives better performance (with the ALE is only 0.29 (m)) than three other algorithms. While in the case of the EPCS algorithm, the MB has to travel along the longest distance to detect location of the sensor nodes. Therefore, the ALE of the EPCS algorithm (ALE = 2.24 (m)) is higher than the ALE of any other proposed algorithms. The SA algorithm and the SGA algorithm stand in the second and the third in the range with the ALE is 0.46 (m) and 0.64 (m), respectively.

In order to compare the performance of the proposed algorithms and some typical methods, a large simulation dataset is created. The results of my proposed algorithms are compared with the algorithm in [143]. The results are summarized in Table 6.2 and are depicted in Figure 6.9.

Figure 6.9 depicts ALE and AET versus the velocity of the MB. In figure 6.9a, it is clear that all algorithms improved their positioning accuracy when the velocity of MB decreases. With the range of velocity of the MB is from 5(m/s) to 50(m/s), the ALE of the SAS algorithm slightly increases from 0.26 to 0.78 (m), this range of the SA algorithm is from 0.41(m) to 0.95(m), but they are almost lower than that of the algorithm in
Figure 6.8: The location error of each sensor node

It is noteworthy that, if the velocity of the MB is lower than 12(m/s), three of the investigated algorithms (i.e., the SGA algorithm, the SA algorithm and the SAS algorithm) achieve better performance than the algorithm in [143]. In figure 6.9b, as the speed of the MB increases, the AET values decrease because with the shorter travel time, the MB will get more messages from its neighbor nodes for localization. The MB in the EPCS algorithm needs to move on the long distance of the EPC to locate the positions of sensors, while the MBs in three other algorithms (i.e., SGA algorithm, SA algorithm and SAS algorithm) only choose the shortest path with the steepest ascent of the RSS. As a result, the AET values of the SGA algorithm and the SAS algorithm are smaller than that of the EPCS algorithm. And they also are lower than the AET of the algorithm in [143]. Comparing to the algorithm in [143], the SGA algorithm finishes its localization tasks faster 4.6 (s) and this value is 3(s) with the SAS algorithm. However, the SA algorithm and the EPCS algorithm have the longer AET, 1.5 (s) and 4.5 (s) than that by the algorithm in [143], respectively.
Table 6.2: The performance of localization algorithms based on the ALE and the AET

<table>
<thead>
<tr>
<th>v (m/s)</th>
<th>Approach [143]</th>
<th>EPCS</th>
<th>SAS</th>
<th>SGA</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ALE</td>
<td>AET</td>
<td>ALE</td>
<td>AET</td>
<td>ALE</td>
</tr>
<tr>
<td>10</td>
<td>0.74</td>
<td>17.86</td>
<td>1.82</td>
<td>20.98</td>
<td>0.26</td>
</tr>
<tr>
<td>20</td>
<td>0.75</td>
<td>8.99</td>
<td>3.0</td>
<td>15.2</td>
<td>0.38</td>
</tr>
<tr>
<td>30</td>
<td>0.74</td>
<td>6.85</td>
<td>3.85</td>
<td>12.2</td>
<td>0.48</td>
</tr>
<tr>
<td>40</td>
<td>0.73</td>
<td>4.81</td>
<td>4.8</td>
<td>9.2</td>
<td>0.61</td>
</tr>
<tr>
<td>50</td>
<td>0.74</td>
<td>4.35</td>
<td>5.65</td>
<td>7.8</td>
<td>0.78</td>
</tr>
</tbody>
</table>

6.6 Conclusions and directions for future research

In this chapter, my proposed algorithms for locating and tracking the unknown position sensor nodes, which are based on the RSS received at the MB, are presented. It is found that the disadvantages of the RSS methods on the NLoS environments can be reduced significantly by the movement of the MB. It also can be a good potential method for localization on the real environments with moving obstacles. Furthermore, all computations needed for localization problem are performed on the MB so it is more power efficient. It is shown that the proposed SAS algorithm and SGA algorithm are able to locate the positions of sensor nodes with a high degree of accuracy despite multi-path effects, fading and shadowing in NLoS environments. Compared with some previous localization algorithms, my proposed algorithms not only improve the localization accuracy but also decrease the execution time when the velocity of the MB is lower than 12(m/s).
Figure 6.9: The ALE and AET versus the velocity of the MB
Chapter 7

SUMMARY OF RESULTS AND THESES

7.1 Summary

This dissertation focuses on two important areas of wireless sensor network: energy efficient and reliable communication in WSNs. Each of these problems is of crucial importance in present and future trends of wireless sensor networks. Therefore, my algorithms focus on optimizing the trade-off between energy efficiency and reliability in data transmission. As a consequence, my algorithms achieve high energy efficiency while still maintaining high reliability in data transmission. My proposed protocols in WSNs have a wide range of applications. By exploring polynomial complexes in WSN communication, I have provided some fast and reliable solutions which outperform other benchmark methods. In both major requirements (i.e., energy efficiency and reliability) of WSNs, I have managed to come up with novel approaches which:

- achieve exact mathematical formulation of objective functions for routing solutions in WSNs;
- have a survey and a complete quantitative comparison of numerical methods for energy-efficient and reliable routing protocols in WSNs;
- achieve energy-efficient and reliable routing protocols in WSNs;
- are proved to be highly flexible even under fast-changing environments;
- have low computation complexity of the proposed algorithms and they can be executed in polynomial order of time;
- reduce the runtime significantly by executing the multicarrier selection in parallel manners.

Furthermore, I managed to make a number of other contributions to the solution of each problem. In case of resource management and packet scheduling, the proposed scheduling algorithm achieves the new efficient method to transmit data with low energy consumption, and with high transmission reliability by developing a smart scheduler. In maximizing the network lifetime in MWSNs, I developed a new method for gathering data in short time with small energy consumption. In improving the QoS routing for WSNs, I developed a
routing algorithm which can optimize multi-path routing in WSNs and improves the energy efficiency under a reliability constraint. In outlier detection problem, I have proven that my proposed algorithm can be used to detect the network violations. I also developed Simulated Annealing method for sensor node localization, which has low average location error and execution time.

Considering the above results, I have achieved the aims of the dissertation. Finally, in each case, I implemented a proof of concept and have run extension simulations on both synthetic and real-world data. My proposed methods and algorithms may be used in scheduling of computational resources, monitoring applications, detecting network violations, tracking applications, robotic strategies, and so on. The applications of these are summarized in Table 7.1.

**Table 7.1:** Summary of my theses

<table>
<thead>
<tr>
<th>A.1. Field: Resource management and packet scheduling</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.2. Performance</td>
</tr>
<tr>
<td><strong>Characteristic</strong></td>
</tr>
<tr>
<td>Improvement in system cost (%)</td>
</tr>
<tr>
<td>System reliability (with $M = 29$)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A.3. Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSNs</td>
</tr>
<tr>
<td>Packet scheduling, LAN protocols</td>
</tr>
<tr>
<td><strong>Other areas</strong></td>
</tr>
<tr>
<td>Telecommunication, Resources management, the resource may include in human resource, financial resources, human skills, production resources, or some natural resources.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B.1. Field: Maximizing the network lifetime in MSWNs</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.2. Performance</td>
</tr>
<tr>
<td><strong>Characteristic</strong></td>
</tr>
<tr>
<td>Network lifetime</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B.3. Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSNs</td>
</tr>
<tr>
<td>Cluster head election in WSNs, efficient mobility technology in WSNs</td>
</tr>
<tr>
<td><strong>Other areas</strong></td>
</tr>
<tr>
<td>Optimize movement schedule, Robotic, Traveling salesman problem applications</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C.1. Field: Routing protocol for WSNs</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.2. Performance</td>
</tr>
<tr>
<td><strong>Characteristic</strong></td>
</tr>
<tr>
<td>Network lifetime</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Network reliability</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C.3. Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSNs</td>
</tr>
<tr>
<td>Efficient routing technique for WSN</td>
</tr>
<tr>
<td><strong>Other areas</strong></td>
</tr>
<tr>
<td>Quality of Service in telecommunication networks</td>
</tr>
</tbody>
</table>

*Continued on next page*
Table 7.1 – Continued from previous page

D.1. Field: Outlier detection in WSNs data

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Method</th>
<th>Achieved value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification rate</td>
<td>Existing method</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td>Novel</td>
<td>0.78</td>
</tr>
</tbody>
</table>

D.2. Performance

D.3. Applications

**WSNs**
Detecting outlier values and events in sensor readings

**Other areas**
Detecting network violations, outlier detection in some realistic monitor applications

E.1. Field: Position location technique

E.2. Performance

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Method</th>
<th>Achieved value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average location error</td>
<td>Existing method</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>SGA</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td>SA</td>
<td>0.41</td>
</tr>
<tr>
<td>Average execution time</td>
<td>Existing method</td>
<td>17.86</td>
</tr>
<tr>
<td></td>
<td>SGA</td>
<td>8.8</td>
</tr>
<tr>
<td></td>
<td>SA</td>
<td>17.8</td>
</tr>
</tbody>
</table>

E.3. Applications

**WSNs**
Tracking sensor location, find the best routing for WSNs based on location of nodes.

**Other areas**
Traffic tracking applications, robotic strategies.

7.2 Future research plan

Although energy efficient and reliable communication in WSNs have been well studied in the last decade, there are still many objectives that must be accomplished before producing realistic WSN applications. In this section, I briefly mention some areas of future work based on my thesis to design an energy-efficient and reliable routing protocol in WSNs.

- Creating and using a new routing protocol for WSNs, which guarantees some critical performance parameters such as latency, throughput, energy consumption, error rate, security and privacy.
- WSN has some limited resources and capacities (e.g., power, bandwidth, storage capacity), as future work I plan to improve the balancing between QoS requirements and energy consumption levels in routing protocol.
- Exploiting mobility has been one of the most concerned issues in WSNs. In my thesis, I have utilized a moving beacon to collect sensed data and detect localization of Nodes in sensing field. These research have opened numerous research directions, which may be further explored. However, mobility has also its advantages and disadvantages. Depending on the purposes and requirements of each application, we
can choose the most suitable mobility for WSN designing. Therefore, in this section, I will capture some limitations that mobility may have, and then propose some possible heuristic solutions to overcome these limitations. These heuristic solutions are described in Table 7.2.

**Table 7.2: Heuristic solutions to overcome the limitations of mobility models**

<table>
<thead>
<tr>
<th>Limitations</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>For some practical reasons, wireless sensor node should be tinny in size. However, its battery size is also small and it is constrained in energy power</td>
<td>To improve the network lifetime with the limitation of energy power, we can choose one of following techniques:</td>
</tr>
<tr>
<td></td>
<td>• Using suitable materials for batteries [110], which have a higher power density, eco-friendly, lighter in weight, lower self-discharge, quicker charging, and a longer lifespan with thousands of charge-discharge cycles.</td>
</tr>
<tr>
<td></td>
<td>• Using materials for renewable and sustainable energy [102].</td>
</tr>
<tr>
<td></td>
<td>• Improving network lifetime by using wireless charging technology [91]. It is observed that most of the existing schemes exploit mobility to prolong the network lifetime while few authors focus on combining mobility and the efficient wireless charging to improve the network lifetime. In the facts, the wireless charging is more convenient than traditional wired charging methods, which helps to charge the wireless sensor nodes automatically without stopping the system. We believe that the mobility coupled with the efficient wireless charging in WSNs will improve the network lifetime significantly.</td>
</tr>
<tr>
<td></td>
<td>• Improving energy capacity, reducing energy consumption by some Energy-Efficient routing techniques [8, 82], utilizing some density deployment methods [163], and smart cluster head election approaches [32].</td>
</tr>
</tbody>
</table>

Different with a static network, in a mobility network, the locations of mobile nodes are changed over time. Therefore, it has to spend more energy and time for localization tasks. After a period of running time, mobile nodes can return to the support center for recharging themselves. Therefore, the amount of energy spent for these mobile nodes’ activities does not affect the network lifetime. In some other cases (in the battlefield), when mobile nodes cannot come back to the support center, they can alternatively do their tasks with flexible wakeup/sleep scheduling for mobiles nodes.
Dynamic topology is also a big problem of the mobility network. When the mobile nodes move the communication between them and source nodes also changes. It becomes very quickly out of date due to these changes. Storage overflow may occur if we do not have reasonable solutions. Therefore, an energy–efficient clustering approach for capturing unexpected events in WSN (as proposed in [32]) is the best solution for these situations.

In mobility models, direction movement, and velocity of mobile sensor nodes are very important [66]. They affect directly the lifetime and QoS of the network. To solve this problem, we can combine between the speed of mobile nodes and their transmission ranges to make the best schedule for movement and transmission tasks of mobiles sensor nodes.
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOA</td>
<td>Angle of Arrival</td>
</tr>
<tr>
<td>AR</td>
<td>Auto-Regressive</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>CH</td>
<td>Cluster-Head Node</td>
</tr>
<tr>
<td>CNs</td>
<td>Cluster member Nodes</td>
</tr>
<tr>
<td>DDoS</td>
<td>Distributed Denial of Service</td>
</tr>
<tr>
<td>DOA</td>
<td>Direction of Arrival</td>
</tr>
<tr>
<td>DoS</td>
<td>Denial-of-Service</td>
</tr>
<tr>
<td>FDMA</td>
<td>Frequency Division Multiple Access</td>
</tr>
<tr>
<td>ICMP</td>
<td>Internet Control Message Protocol</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>ISI</td>
<td>Inter Symbol Interference</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>MAD</td>
<td>Median Absolute Deviation</td>
</tr>
<tr>
<td>MB</td>
<td>Moving Beacon</td>
</tr>
<tr>
<td>MS</td>
<td>Mobile Sink</td>
</tr>
<tr>
<td>MWSNs</td>
<td>Mobile Wireless Sensor Networks</td>
</tr>
<tr>
<td>NCC</td>
<td>Network Control Center</td>
</tr>
<tr>
<td>NLoS</td>
<td>Non-Line-of-Sight</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing</td>
</tr>
<tr>
<td>Q.E.D</td>
<td>What was to be demonstrated</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RSSI</td>
<td>Received Signal Strength Indicator</td>
</tr>
<tr>
<td>SN</td>
<td>Source Node</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>TDOA</td>
<td>Time Difference of Arrival</td>
</tr>
<tr>
<td>TOA</td>
<td>Time of Arrival</td>
</tr>
<tr>
<td>TSP</td>
<td>Traveling Salesman Problem</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>WSNs</td>
<td>Wireless Sensor Networks</td>
</tr>
</tbody>
</table>
Bibliography


Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.


Appendix

A.1 Proof of Lemma 1 in section 6.4 [35]

Assuming that \((\bar{x}, \bar{y})\) is one cluster point of the sequence of the MB’s positions at the iteration \(k\): \(\{x(k), y(k)\}\). Using the definition of RSS at the position of one sensor node in (6.4) we have \(\text{RSS}(\bar{x}, \bar{y}) - \text{RSS}_{\text{threshold}} \leq \delta\). So the proof of this lemma only focuses on the equation:

\[ \nabla f(\text{RSS}(\bar{x}, \bar{y})) = 0. \]

With then suppose \(f(\text{RSS}(x, y)) : \mathbb{R}^n \rightarrow \mathbb{R}\) is continuously differentiable on the set \(S = \{\text{RSS} \in \mathbb{R}^n | f(\text{RSS}(x, y)) \leq f(\text{RSS}(x(0), y(0))) \}\) and that \(S\) is a closed and bounded set. At least one cluster point of the sequence \(\{x(k), y(k)\}\) must exit. Without loss of generality, assuming that \(\lim_{k \to \infty} (x(k), y(k)) = (\bar{x}, \bar{y})\), that

\[ \nabla f(\text{RSS}(\bar{x}, \bar{y})) \neq 0. \]

This being the case, there is a value of \(\bar{\alpha} > 0\) such that \(\delta := f(\text{RSS}(\bar{x}, \bar{y})) - f(\text{RSS}(\bar{x}, \bar{y}) + \bar{\alpha} \bar{d})\), where \(\bar{d} = -\nabla f(\text{RSS}(\bar{x}, \bar{y}))\) then also \(\text{RSS}(\bar{x}, \bar{y}) + \bar{\alpha} \bar{d} \in \text{int}S\). Now \(\lim_{k \to \infty} (d^k) = \bar{d}\) then since \(\text{RSS}(\bar{x}, \bar{y}) + \bar{\alpha} \bar{d} \in \text{int}S\) and \(f(\text{RSS}(x(k), y(k))) + \bar{\alpha} d^k) \rightarrow \text{RSS}(\bar{x}, \bar{y}) + \bar{\alpha} \bar{d}\) sufficiently large resulting in

\[ f(\text{RSS}(x(k), y(k)) + \bar{\alpha} d^k) \leq f(\text{RSS}(\bar{x}, \bar{y}) + \bar{\alpha} \bar{d}) + \frac{\delta}{2}. \]

However,

\[ f(\text{RSS}(\bar{x}, \bar{y})) < f(\text{RSS}(x(k), y(k))) + \alpha^k d^k) \]

\[ \leq f(\text{RSS}(x(k), y(k))) + \bar{\alpha} d^k) \]

\[ \leq f(\text{RSS}(\bar{x}, \bar{y})) - \frac{\delta}{2}. \]

which is of course a contradiction. Thus \(\bar{d} = \nabla f(\text{RSS}(\bar{x}, \bar{y})) = 0.\)  

A.2 Proof of \(\mathcal{R}_{\text{c}}(t) \leq \mathcal{R}_{\text{max}}\)

We assume that \(O_i\) and \(H_i\) \((i = 1, ..., 6)\) denote the best palace for the MS to collect data from \(CH_i\) before and after changing the transmission ranges of the CHs, respectively. As given in Figure A.2.1, our objective is to proof:

\[ H_1H_2 + H_2H_3 + H_3H_6 + H_6H_5 + H_5H_4 + H_4H_1 \leq O_1O_2 + O_2O_3 + O_3O_6 + O_6O_5 + O_5O_4 + O_4O_1. \]

Let \(U_1\) and \(Q_1\) be the points of intersection of \(\overline{O_1O_2}, \overline{O_1O_3}\) and the circle whose center is at \(O_1\) and radius \(r_1\); \(U_2\) and \(Q_2\) the points of intersection of \(\overline{O_4O_5}, \overline{O_1O_4}\) and the circle whose center is at \(O_4\) and radius \(r_4\). Then we need to proof that \(H_1H_2 \leq O_1O_1\), where \(H_1\) and \(H_4\) are midpoints of segment \(U_1Q_1\) and \(U_2Q_2\), respectively.

Considering right triangle \(\triangle Q_1J_1H_1\), we have \(J_1H_1 < Q_1H_1\). While considering the isosceles triangle \(\triangle Q_1O_1U_1\), we have \(Q_1H_1 \leq Q_1O_1 = r_1\). Therefore, \(J_1H_1 < Q_1H_1 \leq r_1\). Similarly, considering right triangle \(\triangle Q_2J_2H_4\), we have \(J_2H_4 \leq r_4\). Considering the
trapezoid $J_1J_2Q_2Q_1$, we have $\overline{J_1J_2} \leq \overline{Q_1Q_2}$. Therefore,

$$
\overline{H_1H_4} = \overline{H_1J_1 + J_1J_2 + J_2H_4} \\
\leq \overline{O_1Q_1 + Q_1Q_2 + Q_2O_4} \\
\leq \overline{O_1O_4} \tag{A.2.1}
$$

In a very similar manner one can obtain

$$
\begin{align*}
H_1H_4 & \leq O_1O_2 \\
H_2H_5 & \leq O_2O_3 \\
H_3H_6 & \leq O_3O_6 \\
H_6H_5 & \leq O_6O_5 \\
H_5H_4 & \leq O_5O_4 \\
H_1H_2 & \leq O_4O_2 \\
H_4H_1 & \leq O_4O_1 \\
\end{align*} \tag{A.2.2}
$$

It results that $\Re_s(t) \leq \Re_{\text{max}}$