Mitigating energy holes in wireless sensor actuator networks using a mobile sink and charger

Hamidreza Salarian
University of Wollongong

Recommended Citation
UNIVERSITY OF WOLLONGONG

COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
Mitigating Energy Holes in Wireless Sensor Actuator Networks

Using a Mobile Sink and Charger

A thesis submitted in fulfillment of the requirements for the award of the degree

Doctor of Philosophy

from

University of Wollongong

by

Hamidreza Salarian
School of Electrical, Computer and Telecommunications Engineering

August 2013
In memory of my father
Abstract

Wireless Sensor Networks (WSNs) are comprised of miniature, low-cost, low-powered, multi-functional sensor nodes with the ability to measure various parameters associated with their environment. In particular, they are able to monitor tremor, distance, direction, speed, load and pressure, temperature, humidity, light, vibration, motion and sound. Recently, researchers have extended the capability of WSNs to create Wireless Sensor Actuator Networks (WSANs), where nodes have the ability to manipulate their environment using actuators. Every node in both WSNs and WSANs has the ability to collect and process sensed data, and forward it to one or more sink/actuator nodes via its wireless transceiver in a multi-hop manner. Multi-hop communications, however, result in nodes that are close to a sink or an actuator to have a faster energy dissipation. This leads to non-uniform energy depletion, and the formation of energy holes around the sink / actuator.

This thesis studies and proposes solutions that mitigate the formation of energy holes. In particular, it explores the use of mobile sinks and proposes two innovative scheduling algorithms to generate the trajectory of a mobile sink. In the first method, a heuristic approach called Weighted Rendezvous Planning (WRP) is developed in which each sensor node is assigned a weight corresponding to its hop distance from the tour and the number of data packets that it forwards to the closest Rendezvous Point (RP). Experimental
results indicate that WRP enables a mobile sink to retrieve all sensed data within a given deadline whilst conserving the energy consumption of sensor nodes. The results show that WRP can reduce energy consumption by 22% and increase network lifetime by 44% in comparison with existing algorithms.

The second method considers a mobile rover/robot with wireless recharging capability. The charging problem is formulated as an Integer Linear Program (ILP) with the objective of maximizing network lifetime. The obtained model is equivalent to the well known NP-hard, Capacitated Vehicle Routing Problem (CVRP). A heuristic method called Binary Search Wireless Charging (BSWC) is then developed in which the mobile charger preferentially visits sensor nodes with the shortest lifetime. BSWC uses the binary search algorithm to find the target lifetime that minimizes the residual energy of the rover’s battery as well as using the shortest Hamiltonian path to reduce travelling cost. BSWC is validated mathematically and sufficient conditions are derived to ensure infinite network lifetime. Simulation results demonstrate that BSWC increases network lifetime by 400% as compared to Greedy-Plus, the current state-of-the-art algorithm.
Statement of Originality

I, Hamidreza Salarian, declare that this thesis, submitted in fulfillment of the requirements for the award of the degree Doctor of Philosophy in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work is entirely my own work, except where due reference is made in the text.

No work in this thesis has been submitted for a degree to any other university or institution.

Signed

Hamidreza Salarian
August 2013
Acknowledgements

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis.

First and foremost, I would like to express my deepest appreciation to my supervisors Professor Fazel Naghdy and A/Professor Kwan-Wu Chin for their guidance, patience and support throughout this research and the writing of this thesis. Their insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. Without their guidance and persistent help this dissertation would not have been possible.

I also like to thank all reviewers at journals for their time and efforts they spent in giving valuable feedbacks. Their suggestions have certainly raised the quality of my publications.

Furthermore, I would like to thank my family: my mother, my wife, my brothers and their lovely family. They were always supporting me and encouraging me with their best wishes.

Last but not least, a thank you to Professor Tim Marchant the Dean of Research, who has supported our team throughout entire process.
Contents

1. Introduction .......................................................... 1
   1.1 Overview ...................................................... 1
   1.2 Wireless Sensor Networks ................................. 2
   1.3 Wireless Sensor / Actuator Networks .................... 3
   1.4 Problem Statement ........................................... 6
   1.5 Research Hypothesis, Aim and Objectives ............... 7
   1.6 Research Approach .......................................... 9
   1.7 Contributions ................................................. 12
   1.8 Thesis Structure ............................................. 13

2. Literature Review .................................................. 15
   2.1 Introduction ................................................... 15
   2.2 Coordination of Wireless Sensor Actuators .......... 16
      2.2.1 WSAN Architecture ................................. 18
      2.2.2 Coordination ........................................... 21
      2.2.3 Sensor-Actuator Coordination ..................... 26
      2.2.4 Routing Protocols .................................... 30
      2.2.5 Transport Protocols .................................. 36
      2.2.6 Actuator-to-Actuator Coordination ............... 40
      2.2.7 Summary ................................................. 44
2.3 Mobile Sink .................................................. 45
  2.3.1 Introduction ............................................. 45
  2.3.2 Real-Time Applications ................................. 48
  2.3.3 Delay-Tolerant Applications ............................ 52
    2.3.3.1 Direct ........................................... 52
    2.3.3.2 Rendezvous ...................................... 54
    2.3.3.3 Fixed .......................................... 54
    2.3.3.4 Unconstrained .................................. 55
  2.3.4 Discussion ............................................... 58
2.4 Mobile Charger .............................................. 59
  2.4.1 Introduction ............................................. 59
  2.4.2 Unlimited Battery Capacity .............................. 62
  2.4.3 Limited Battery Capacity ................................. 63
2.5 Summary ..................................................... 67

3. Weighted Rendezvous Planning ................................. 69
  3.1 Introduction ............................................... 69
  3.2 Problem Formulation ....................................... 69
    3.2.1 Assumptions ......................................... 70
    3.2.2 Notation ............................................ 71
    3.2.3 Delay-aware Energy Efficient Travel Path (DEETP) ... 73
  3.3 Weighted Rendezvous Planning .............................. 74
    3.3.1 Analysis ............................................. 79
  3.4 Validation ................................................. 83
    3.4.1 Performance under SPT ............................... 86
List of Figures

1.1 A block diagram of a typical sensor architecture. . . . . . . . . 2
1.2 A typical WSN architecture. . . . . . . . . . . . . . . . . . . . 3
1.3 A block diagram of a typical actuator architecture. . . . . . 4
1.4 A wireless sensor actuator network. . . . . . . . . . . . . . . . 5
1.5 A typical wireless sensor network with random uniform sensor
    node distribution. . . . . . . . . . . . . . . . . . . . . . . . . . 6

2.1 Two architectures of WSANs: (a) fully-automated and (b)
    semi-automated architectures. . . . . . . . . . . . . . . . . . . 19
2.2 Sensor-actuator coordination mechanism: a) single and b)
    multiple actuators. . . . . . . . . . . . . . . . . . . . . . . . . 22
2.3 Actuator-actuator coordination mechanism: a) centralized, b)
    distributed. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
2.4 An example showing a mobile sink performing data collection
    in a WSN. A source node determines and sends all data to a
    suitable site. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
2.5 A hybrid movement pattern for a mobile sink node. Source
    nodes generate and send sensed data to the nearest RP. Each
    RP buffers received data and waits until the mobile sink arrives. 47
2.6 Tour finding steps in CB. . . . . . . . . . . . . . . . . . . . . . . 58
2.7 A mobile wireless charger in a WSN. .......................... 61

3.1 Example of WRP operating in a WSN with 10 nodes. .... 78
3.2 Network energy consumption between brute force and WRP. 87
3.3 Standard deviation of sensor nodes’ energy consumption rate. 88
3.4 Network energy consumption for WRP, CB, RD-VT and RP-
     UG. ................................................................. 89
3.5 Standard deviation of sensor nodes’ energy consumption for
     WRP, CB, RD-VT and RP-UG models .......................... 90
3.6 Network lifetime for WRP, CB, RD-VT and RP-UG. ......... 91
3.7 Network energy consumption for WRP, CB, RD-VT and RP-
     UG under different required delivery times for data packets. .... 92
3.8 Network lifetime for WRP, CB, RD-VT and RP-UG under
     different required delivery times for data packets. .......... 93
3.9 Simulation time for WRP, CB, RD-VT and RP-UG. ......... 94
3.10 Network energy consumption for WRP, CB, RD-VT and RP-
     UG in SMT scenarios. ........................................ 95
3.11 Standard deviation of sensor nodes’ energy consumption in
     SMT scenarios. ................................................. 97
3.12 Network lifetime for WRP, CB, RD-VT and RP-UG in SMT
     scenarios. ...................................................... 98

4.1 A simple WSN with three nodes. ................................. 105
4.2 Line topology. .................................................. 111
4.3 A simple WSN with three sensor nodes. ....................... 120
4.4 \( d_{\text{max}} \) for different network sizes. .................... 123
4.5 Mobile charger’s battery capacity for different network sizes. . 123
4.6 Network lifetime for different value of $K$ . . . . . . . . . . 124
4.7 Second experimental topology with 15 sensor nodes. . . . . . 125
4.8 Achieved lifetime of the network shown in Figure 4.7 for $K = 1$
   to $K = 10$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 126
4.9 Achieved network lifetime for BSWC and GP algorithm in
   networks with 9 and 15 nodes. . . . . . . . . . . . . . . . . . . . . . 127
## List of Tables

1.1 Summary of WSAN applications .................................................. 5

2.1 Comparison of sensor-to-actuator proposals ................................. 30

2.2 Comparison of routing protocols ............................................... 36

2.3 Comparison of transport protocols .............................................. 41

2.4 Comparison of actuator-to-actuator protocols ............................... 44

2.5 Comparison of proposed mobile sink moving patterns for real time applications ......................................................... 51

3.1 A summary of notations used in problem formulation .................. 74

3.2 Simulation parameters ............................................................. 85

4.1 A summary of notation ............................................................. 104

4.2 Scenario’s parameters ............................................................. 111

4.3 Simulation parameters ............................................................. 122
Chapter 1

Introduction

1.1 Overview

Wireless Sensor Networks (WSNs) and its extension Wireless Sensor Actuator Networks (WSANs) consist of smart, battery powered, wireless devices called sensor nodes. These nodes self-configure and self-organize to form a sensing and data forwarding platform capable of measuring environment parameters, and in the case of WSANs, nodes are able to affect their environment. Sensed data are usually processed and forwarded to one or more sink / actuator nodes in a multi-hop manner. In addition, the sink / actuator nodes may issue commands to control the wake-up schedule or actuation of sensor nodes. A fundamental problem that arises when forwarding data is that sensor nodes close to a sink or an actuator experience faster energy dissipation. This results in non-uniform energy depletion, and formation of energy holes around the sink / actuator. To this end, this thesis will investigate the use of a mobile sink / charger to prevent the formation of energy holes.

This chapter will initially provide an overview of the basic characteristics of WSNs and WSANs. Then it outlines the fundamental problems studied
in this research, followed by research hypothesis, aims and objectives. The research methodology deployed to explore the hypothesis and to address the identified problems will be explained. This will be followed by a summary of the major contributions and outcomes resulting from this thesis. The chapter will conclude with an overview of the thesis structure.

1.2 Wireless Sensor Networks

Advances in micro-electro-mechanical systems (MEMS) technology and wireless communications have enabled the development of small size, low-cost, low-power, multi-functional devices that are typically equipped with sensors such as seismic, magnetic, thermal, visual, infrared, acoustic and radar. Consequently, such sensors can be used to measure tremor, distance, direction, speed, load and pressure, temperature, humidity, light, vibration, motion and acoustic parameters. The structure of a typical sensor node is illustrated in Figure 1.1. A sensor node usually consists of a power supply, transceiver, memory, processing unit, and sensing unit. The data captured by a sensor is converted to a digital signal through an analog to digital converter (ADC) unit. The processor collects and processes the signal and sends it to the transmission unit [1] [2].

A collection of sensor nodes can be configured as a WSN with wide rang-
1. INTRODUCTION

Fig. 1.2: A typical WSN architecture.

ing applications, including precision agriculture [3][4][5], pest monitoring [6], and volcanoes monitoring [7] to name a few. Figure 1.2 shows a typical WSN, comprising of one sink (or base station) and a number of sensor nodes scattered in a geographical space randomly or according to a predefined structure. A key characteristic of WSNs is that each node can forward its data to the sink node, usually connected to the Internet via a gateway.

1.3 Wireless Sensor / Actuator Networks

Recently, researchers have equipped sensor nodes with an actuator to enhance their functionalities. An actuator is a transducer that converts electric signal to linear or angular motion. Valves that control the water or gas outflow from a pipe, electrical motors that open/close doors and windows, and switches that turn on/off heaters or lights are some examples of actuators. Figure 1.3 shows the main components of an actuator node; i.e., transmission, processor, storage, controller (decision), Digital to Analog Converter (DAC) and actuation unit. The decision unit generates an action command according to the sensory information that it receives. For example, the ME8300 Wireless Zone Valve actuator from Spartan [8] consists of a TransCeivers Module
1. INTRODUCTION

Fig. 1.3: A block diagram of a typical actuator architecture.

(TCM) [9], 16MHz 8051 CPU, 32KB Flash and 2kB RAM, 8bit DAC, and 868 MHz/315 MHz transceiver, Alternating Current (AC) motor as an actuation unit and 24 volt AC power supply.

The resulting sensor nodes thus have the capability to act on the environment through one or more actuators. A distinguishing feature of actuator nodes is that they are usually resource-rich, compared to sensor nodes. Moreover, they have high power consumption and computational power, and have a longer communication range. For these reasons, in a given environment, there are fewer actuators than sensors [10].

Figure 1.4 shows a typical WSAN, comprising of geographically distributed network of wireless sensor and actuator nodes. Similar to WSNs, sensor nodes transmit the collected data to actuator nodes via single-hop or multi-hop transmissions. The actuator nodes then decide on an appropriate response to manipulate the sensor field. In such a network, sink nodes provide the end user with the ability to communicate with sensor and actuator nodes, monitor their operation and manage them.

The characteristics of various applications of WSANs are outlined in Table 1.1. Briefly, these applications include home automation [11][12][13], animal control [14], infrastructure health monitoring [15] [16] [17] [18] [19], precision agriculture [20][5][21][22], intelligent buildings [23][24][25][26], sewer overflow management [27][28] and traffic light control [29][30]. The number
1. INTRODUCTION

Fig. 1.4: A wireless sensor actuator network.

<table>
<thead>
<tr>
<th>Applications</th>
<th>Sensor Types</th>
<th>Actuator Types</th>
<th>Delay Tolerance</th>
<th>Number of Sensors</th>
<th>Number of Actuators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home Automation</td>
<td>Temperature, humidity, light, acoustic and occupancy sensors.</td>
<td>Relays, light, switches, motors, noise-masking system adjusting and valves.</td>
<td>2 seconds</td>
<td>More than 100 sensor nodes</td>
<td>20-50 actuator nodes</td>
</tr>
<tr>
<td>Animal control</td>
<td>Location and velocity meter sensors</td>
<td>Stimulate board</td>
<td>800 milliseconds</td>
<td>20-30 sensor nodes</td>
<td>20-30 actuator nodes</td>
</tr>
<tr>
<td>Infrastructure health monitoring</td>
<td>Velocity meter and piezoelectric sensors</td>
<td>Hydraulic or piezoelectric shakers</td>
<td>1 Day</td>
<td>20-100 sensor nodes</td>
<td>20-100 actuator nodes</td>
</tr>
<tr>
<td>Precision agriculture</td>
<td>Temperature, humidity, light and soil moisture sensors</td>
<td>Valves, relays light switches</td>
<td>10 minutes</td>
<td>More than 100 sensor nodes</td>
<td>20-30 actuator nodes</td>
</tr>
<tr>
<td>Intelligent Buildings</td>
<td>Smoke detectors, glass break and motion sensors</td>
<td>Water sprinklers and cameras</td>
<td>2 seconds</td>
<td>20-50 sensor nodes</td>
<td>20-30 actuator nodes</td>
</tr>
<tr>
<td>Sewer management</td>
<td>Water level</td>
<td>Valves or city sewers</td>
<td>14 minutes</td>
<td>More than 100 sensor nodes</td>
<td>More than 100 actuator nodes</td>
</tr>
<tr>
<td>Traffic light control</td>
<td>Magnetic sensor</td>
<td>Relays of traffic lamp</td>
<td>10 seconds</td>
<td>More than 100 sensor nodes</td>
<td>More than 100 actuator nodes</td>
</tr>
</tbody>
</table>

Tab. 1.1: Summary of WSAN applications

of sensor and actuator nodes in each application varies according to the nature of application. For example, in sewer management system and traffic light control, due to the large coverage area, the number of sensors and actuators may be in the order of hundreds as compared to a few nodes when they are used to monitor buildings. In home automation and precision agriculture, there are more sensor nodes than actuator nodes due to the large coverage area. In animal control and infrastructure health monitoring applications, sensor and actuator nodes are integrated and hence are present in equal numbers.
1.4 Problem Statement

As mentioned, a sensor node has the capability to collect and process data, as well as forward the data to a sink or an actuator node via its wireless transceiver in a multi-hop manner. As a result, nodes that are in the vicinity of a sink or an actuator tend to become congested as they are responsible for forwarding data from nodes that are further away. Thus, the closer a sensor node is to a sink or an actuator, the faster its battery dissipates. On the contrary, sensor nodes located far from a sink or an actuator may have more than 90% of their initial energy during the same period [31]. This leads to non-uniform depletion of energy and results in network partition due to the formation of energy holes [32] [33]. As a result, the lifetime of a WSN / WSAN is bounded by the total energy of sensor nodes in the vicinity of a sink or an actuator node. The energy holes can be repaired by replacing the battery of sensor nodes. However, this is impractical given the large number of sensor nodes and they may be deployed in an inaccessible terrain.
In order to illustrate the need for balancing the energy consumption of sensor nodes and to prevent the formation of energy holes, consider Figure 1.5. The sensor nodes are distributed in a circular area and the sink node is located at the center. The area is divided into three concentric circles: $C_1$, $C_2$ and $C_3$ based on sensor nodes’ transmission range $r$. As highlighted, $C_1$ consists of 10 sensor nodes that are only one hop away from the sink node, $C_2$ includes 20 sensor nodes that are two hops away from the sink, and $C_3$ includes 30 sensor nodes that are three hops away from the sink. Each sensor node generates one data packet at each time interval $T$. As a result, sensor nodes in $C_1$ receive 50 packets from the nodes located in other circles over $T$. This will require each sensor node in $C_1$ to forward an average of five data packets to other sensor nodes. During the same time period, sensor nodes in $C_2$ receive 30 packets from nodes in $C_3$. Thus the average number of data packets forwarded by each sensor node in $C_2$ is 1.5, which is 3.3 times less than sensor nodes in $C_1$. As a result, the sensor nodes in $C_1$ will run out of energy faster than other nodes. In other words, they will be disconnected from the sink sooner, and thereby, adversely affect the operation of the network.

1.5 **Research Hypothesis, Aim and Objectives**

The primary aim of this research is to develop effective mobility based methods that can prevent the formation of energy holes in WSNs by balancing the energy consumption of sensor nodes. In particular, the research will investigate the application of mobile sinks and mobile chargers to mitigate the
formation of energy holes.

Several studies have demonstrated the benefits of using a mobile sink to mitigate energy holes. However, these benefits are dependent on the path taken by the mobile sink, particularly in delay sensitive applications, as all sensed data must be collected within a given time constraint.

In this study we explore how a hybrid moving pattern in which a mobile sink node only visits Rendezvous Points (RPs) as opposed to all nodes addresses the problem at hand effectively. In particular, sensor nodes not designated as RPs forward their sensed data via multi-hop to the nearest RP. Apart from that, this thesis considers the following requirements:

- Mobile sink visits all RPs within a given delay bound.
- The energy consumption of sensor nodes is minimized.
- The energy consumption of sensor nodes is distributed uniformly.
- The network lifetime is maximized.

The second mobility-based approach examined, i.e., the mobile charger, is assumed to be an autonomous mobile rover / robot with wireless recharging capability that visits sensor nodes and wirelessly recharges their batteries. The use of a mobile charger involves a number of critical issues. Namely, the order of visits made to nodes, the recharging time and the distance travelled by the mobile rover. The ideal solution is to charge all sensor nodes up to their maximum battery capacity. However, there may be a large number of sensor nodes. Moreover, the mobile charger has finite battery capacity.
1. INTRODUCTION

As a result, the approach taken in this thesis is to constrain the path of the mobile charger to only a subset of sensor nodes. This requires the identification of the best subset of sensor nodes that effectively utilize the energy provided by the mobile charger. Accordingly, the following objectives are pursued:

- Minimizing the distance traveled by the mobile charger in order to conserve energy.
- Distributing the mobile charger’s energy uniformly among sensor nodes in order to reduce disparity in the lifetime of sensor nodes.
- Utilizing the left over energy of mobile charger at the end of its tour in order to maximize its charging time.

1.6 Research Approach

As a result of the work conducted, two innovative approaches are developed to achieve the stated aim and objectives. The first approach addresses the Delay-aware Energy Efficient Path Problem (DEETP). The main challenge is to identify the most suitable RPs for a mobile sink in order to minimize the energy consumed by sensor nodes during multi-hop communications whilst meeting a given packet delivery bound. It is shown that the problem is NP-hard. A heuristic algorithm called Weighted Rendezvous Planning (WRP) is proposed to determine the mobile sink trajectory and the set of RPs that optimize the energy consumption of sensor nodes.
WRP assigns a weight to each sensor node corresponding to its hop distance from the tour and the number of data packets that it forwards to the closest RP. Giving priority to sensor nodes that forward data packets from denser parts of a WSN during tour computation is critical as these nodes generate the highest number of packets. This strategy decreases the number of congestion points, and in turn, reduces energy consumption, improves WSN lifetime and mitigates the energy holes problem. Analytical results show that selecting the sensor nodes forwarding the highest number of data packets and having the longest hop distance from the tour can reduce the network energy consumption as compared to other nodes. In addition, WRP considers the hop distance between sensor nodes and RPs as fewer hop counts reduce multi-hop transmissions.

Apart from node density and hop count, when using a Steiner Minimum Tree (SMT), WRP uses virtual RPs in the final tour to increase performance and does not replace them with real sensor nodes. Moreover, in contrast to Cluster-Based (CB) [34] and Rendezvous Design for Variable Tracks (RD-VT) [35] algorithms, WRP is proven to find a tour, if it exists. WRP is evaluated against three recently proposed algorithms: CB [34], RD-VT [35] and Rendezvous Planning Utility-based Greedy (RP-UG) [36]. The results show that WRP achieves 22% more energy savings and 44% better distribution of energy consumption as compared to previously proposed algorithms.

In the second approach, a mobile charger is deployed to charge the rechargeable batteries of sensor nodes. A method called Binary Search Wireless Charging (BSWC) algorithm is proposed in which a mobile wireless charger preferentially visits sensor nodes with the shortest lifetime and replenish their
BSWC sorts sensor nodes based on their lifetime in a non-decreasing order and selects the first $K$ sensor nodes to be charged such that their lifetime is equivalent to that of the $(K+1)$-th sensor node. It also minimizes the energy consumption due to travelling by estimating the shortest Hamiltonian path between $K$ nodes. In addition, BSWC increases the target lifetime up to the maximum achievable target lifetime for the selected nodes. Increasing the target lifetime helps the mobile charger spends its energy to charge sensor nodes instead of returning to the depot with residual energy.

The correctness of BSWC is proven through analytical methods. It is shown that, finding a charging tour is guaranteed if a possible charging sequence exists. Moreover, the results show that in contrast to Greedy-Plus [37], BSWC makes a better use of the mobile charger battery as it results in minimum residual energy when it returns to the depot.

In addition, a relationship based on parameter $K$ and the network size when BSWC is run over multiple charging rounds is defined. Using this relationship, the size of a mobile charger battery for minimum value of $K$ is derived. This ensures that all sensor nodes remain alive perpetually. The simulation results show that BSWC increases the network lifetime by 400%
compared to Greedy-Plus, the current state-of-the-art algorithm.

1.7 Contributions

The key contributions of the thesis can be summarized as follows:

• A review of state-of-the-art in sensor/actuator coordination, covering routing protocols, transport protocols, and actuator-to-actuator coordination protocols.

• An in-depth review of methods for deploying mobile sink and mobile charger to mitigate the energy holes problem, including both direct and RP based approaches.

• Definition and formulation of the DEETP and WCP problems with the objective of minimizing energy consumption by reducing multi-hop transmissions from sensor nodes to RPs, and thereby, increasing the lifetime of all sensor nodes up to the maximum achievable lifetime.

• Development of a novel heuristic mobile sink path selection algorithm, called Weighted Rendezvous Planning (WRP), which produces a near optimal travel tour that minimizes the energy consumption of sensor nodes.

• Development of a novel heuristic method called BSWC in which a near optimal charging tour is derived to increase WSN lifetime.

The study resulted in a number of publications, including:
1. INTRODUCTION


1.8 Thesis Structure

The rest of this thesis is structured as follows.

1. Chapter 2. A review of WSN and WSAN is carried out, which is divided into three major parts. The first part consists of a review of the state-of-the-art in WSANs and techniques carried out to address the fundamental problems faced in this field. More specifically, solutions that address the following problems: (i) sensor / actuator coordination, (ii) routing protocols, (iii) transport protocols, and (iv) actuator-to-actuator coordination protocols. This chapter also contains an extensive qualitative comparison of the key features of these solutions as well as their advantages and disadvantages. The second part focuses on works that employ one or more mobile sinks. The third part novels mobile wireless charger algorithms that boost the energy level of sensor
nodes.

2. Chapter 3. This chapter presents the DEETP problem and shows its NP-hard property. It also introduces a heuristic solution, called WRP, and presents a detailed analysis. The key properties of WRP are highlighted and its correctness is proven mathematically. The chapter ends with a quantitative comparison of WRP against previous works.

3. Chapter 4. This chapter expounds WCP, and shows that it is NP-hard. Then, it presents BSWC, a heuristic method that finds a near optimal charging tour. The chapter continues with an extensive performance analysis, both analytical and simulation, of BSWC as well as comparison against the Greedy-Plus model.

4. Chapter 5. This chapter concludes the thesis and summarizes the key outcomes. It also outlines the future research directions.
2.1 Introduction

Mitigating energy holes and increasing the lifetime of sensor nodes are major research problems in WSNs / WSANs [38]. A number of energy conservation methods such as power-aware routing algorithms [39] [40] [33], clustering of sensor nodes [41] [42] and periodic hibernation [43] [44] have been proposed to increase network lifetime. These approaches have limited effectiveness because they suffer from the problem outlined in Section 1.4. Researchers have also considered replacing the batteries of sensor nodes [45] [46], and equipping sensor nodes with energy harvesting technologies such as solar [47] [48] or wind [49].

Recently, the authors of [50, 51, 52, 53, 54, 55, 56, 57, 36, 58, 59, 60, 61, 62, 63] show that a mobile sink / charger can balance the energy consumption of sensor nodes effectively, and thus increase their lifetime. A mobile sink collects sensed data directly from sensor nodes, and thereby, help sensor nodes save energy that otherwise would be consumed in multi-hop communications. Also a mobile charger equipped with a wireless charger visits sensor nodes and replenishes their batteries up to a certain level to maximize network
This chapter will review WSNs and WSANs through three major parts. In the first part, proposed techniques in the following areas will be studied: (i) sensor-actuator coordination, (ii) routing protocols, (iii) transport protocols, and (iv) actuator-to-actuator coordination protocols. An extensive qualitative comparison of the key features as well as their advantages and dis-advantages will be carried out. In the second part the focus will be on studies that employ one or more mobile sinks for data collection and help to balance the energy consumption of sensor nodes, the more specific topic dealt within this thesis. The third part reviews works that consider mobile wireless chargers, which include trajectory computation, and scheduling of sojourn times.

2.2 Coordination of Wireless Sensor Actuators

WSANs have a myriad of applications, ranging from pacifying bulls to controlling light intensity in home automation. An important aspect of WSANs is coordination. Unlike conventional WSNs, sensor and actuator nodes must work hand-in-hand to collect and forward data, and act on any sensed data collaboratively, promptly and reliably.

A key design parameter of WSAN is the delay tolerance of applications. Here, delay tolerance is defined as the allowed time delay between sensing and actuation. This parameter is important because it governs the time within which actuators must respond to sensed data. For example, when a WSAN is used to monitor an infrastructure, sensor nodes have an extremely
low duty cycle, and hence, are asleep most of the time. They either wake up periodically to test the structure, say once a day at midnight, or may be woken to test the structure immediately after a catastrophic event such as an earthquake, a collision, or a blast [19]. In precision agriculture [21] and sewer management system [27], sampling rates of 10 and 14 minutes are sufficient to control and monitor key parameters. In lighting system applications, a sampling rate of two seconds is sufficient as light intensity has an analogue pattern [12]. Lastly, animal control applications require a sampling interval of 500 milliseconds [14].

In WSAN, any actuation carried out as per the delay tolerance of an application is highly dependent on three processes: distributed collection of data by sensor nodes, forwarding the data to actuator nodes, and cooperative decision by actuator nodes on how to perform the required action. These three processes are referred to as coordination in WSANs. In addition to delay tolerance, energy efficiency is important because each sensor and actuator node is equipped with finite battery capacity.

This section focuses on the coordination problem in WSANs. Specifically, it reviews past studies that improve communications between sensors and actuators, and also between actuators. The critical problems include routing and transportation of sensed data and commands; both of which play a crucial role in the operation of a WSAN as they govern how actuators respond to one or more events. Moreover, any solution must ensure that packet delays are bounded, and packets are delivered reliably as they may contain information used to locate actuator nodes, and also control the movement and responses of actuator nodes.
2. LITERATURE REVIEW

In the next section, an overview of two main WSAN architectures is presented. The type of architecture will have a significant influence on the coordination protocol used to meet application requirements. With this in mind, an overview of the coordination problem in Section 2.2.2 will be followed by solutions proposed for sensor-actuator, actuator-to-actuator, routing and transport problems in Section 2.2.3.

2.2.1 WSAN Architecture

The network architecture of WSANs can be categorized as either fully-automated or semi-automated [1] as shown in Figure 2.1. In the fully-automated architecture, actuator nodes coordinate amongst themselves and decide on a plan of action based on sensed data. In the semi-automated architecture, however, sensor nodes route their data to actuators via a sink. Similar to WSNs, a central entity or sink may be used to collect and process sensed data, and send commands to actuator nodes. Alternatively, the sink node may be used just for monitoring and managing the overall network.

Both architectures have their advantages and disadvantages. A semi-automated architecture is similar to WSNs. In addition to using existing WSNs protocols, due to its centralized property, there is no need for a distributed communication and coordination protocol between sensor and actuator nodes. However, since sensed data is routed through a sink rather than a nearby actuator, communication latency can be significant. Moreover, nodes near the sink will deplete their energy quicker than those further away [1]. On the other hand, in the fully-automated architecture, sensed data is sent to different actuator nodes. Hence, the communication load can be distributed
19

2. LITERATURE REVIEW

Fig. 2.1: Two architectures of WSANs: (a) fully-automated and (b) semi-automated architectures.

evenly amongst actuator nodes, and thereby, extend the lifetime of a WSAN [64].

WSNs are passive, where sensor nodes simply record data and send them to one or more sinks for processing, which may then issue new commands to change the sample rate. However, WSANs are active in that sensed data governs the behavior of actuators and help them to manipulate the environment, which in turn affect the data collected by sensors. In short, there is a close coupling between sensor and actuator nodes. This fundamental difference poses a number of new challenges [1] [65] [64]:

- **Coordination.** The communication processes between sensor and actuator nodes play a critical role in WSANs. In particular, sensed data
must be acted upon quickly by a distributed collection of actuators in
order to mount appropriate actions quickly. This means sensor nodes
need to form low delay and reliable paths to one or more actuator
nodes, and actuator nodes must collaboratively reach a consensus on
the best response to one or more events.

• \textit{Timing}. Each event in a WSAN has a required action time, from when
it happens to when the corresponding action has to be carried out
before the event becomes un-controllable. Hence, real-time data com-
munications and coordination are critical to the operation of WSANs.

• \textit{Reliability}. In order to have correct execution of actions, actuator nodes
need to receive sensed data within a pre-determined time period in
order to reconstruct an event, understand its intensity, location and
coverage, and lastly determine the appropriate number of actuators
that are deployed in response to the event.

• \textit{Mixed Traffic}. Events taking place in a WSAN may require varying re-
action times. Moreover, sensed data will have different real-time prior-
ities. For example, in home automation applications, sensor nodes may
have different capabilities, where a subset of sensor nodes may be used
to measure temperature whilst others are used to detect movements.
In addition, sensed data may be of different length and sampling rate.
As a result, data will have to be handled differently by sensor nodes as
they have different delay and reliability requirements.

• \textit{Mobility}. The mobility nature in WSANs is completely different from
WSNs. Mobile elements in WSNs are designed to save the energy of sensor nodes by collecting data from sensor nodes directly or via rendezvous or cache points [36] [54] [66]. On the other hand, in WSANs, mobile actuator nodes are used to reduce end-to-end packet delay and guarantee task completion times, e.g., in areas with high frequency of events.

There are also common challenges and problems associated with both WSNs and WSANs; including, spectrum management [67] [68], Medium Access Control (MAC) [69] and security [70] [71].

### 2.2.2 Coordination

Coordination is a fundamental problem in WSANs. The exact coordination used, however, is dependent on the network architecture. More specifically, coordination is carried out by the sink in the semi-automated architecture. On the other hand, in the fully-automated architecture, there are two modes of communication: sensor-to-actuator and actuator-to-actuator coordination. In the former, sensor nodes are required to find an appropriate actuator node to send their data. In the latter, the actuator nodes coordinate amongst themselves to deal with sensed data [1][72][73]. In this review, the focus is on fully-automated architecture as existing WSNs protocols can be deployed in the semi-automated architecture. Further details on WSNs protocols are provided in [74] and [13].

In the fully-automated architecture, distributed protocols are needed for sensor-to-actuator coordination. This is an important function as a correct response from actuator nodes cannot be achieved unless sensed data arrives
Fig. 2.2: Sensor-actuator coordination mechanism: a) single and b) multiple actuators.

in a timely manner and is received correctly by the corresponding actuator nodes. This gives rise to the following fundamental question: how do sensor nodes determine the best actuator node(s) to send their data? This is important as it determine the location and intensity of an event before taking action.

Once an event occurs, two alternative methods can be considered when forwarding data. That is, the data may be forwarded to an actuator node (see Figure 2.2 (a)), or multiple actuator nodes (see Figure 2.2 (b)). In the first case, sensor nodes in the event area communicate with each other to find the nearest actuator node(s) that covers the said area and has sufficient energy to carry out the required task. The advantage of this approach is that actuator nodes are excluded from this coordination effort. Instead, coordination is carried out by sensor nodes. As a result, sensor nodes around the event area will deplete their energy faster. In the second case, each sensor node independently selects an actuator node [75]. However, there is no sensor-to-sensor coordination, which may overload a given actuator or cause sensor nodes around an actuator node to deplete their energy quicker.

Once sensor nodes decide on one or more actuators, their next task is to
locate and select suitable actuator nodes. This task is difficult in applications where actuator nodes are mobile. Hence, there is a need for mechanisms that enable actuator nodes to update sensor nodes of their locations, and vice-versa. Moreover, these mechanisms must be energy efficient as continuous tracking of actuator nodes increases the duty cycle of resource constrained sensor nodes.

The next issue to be addressed is establishing one or more routing paths to actuator nodes. The main challenge is constructing one or more paths that meet the delay requirement of applications, and are sufficiently reliable to carry data between sensors and actuator nodes. A key consideration is congestion avoidance, especially in areas surrounding an actuator node. Otherwise, a congestion path will lead to packet loss and increased end-to-end delay. Another issue of importance is route construction and maintenance to one or more actuator nodes. Moreover any unicast or multicast routing protocols must consider the low duty cycle of nodes, and the mobility of actuator nodes.

Actuators must coordinate amongst themselves to ensure that at least one of them responds to any event that arises. Figure 2.3 shows two decision categories of actuator-actuator coordination. In the centralized decision category, whenever an actuator node receives an event from a sensor node, it sends the information to a predetermined, central actuator node or decision centre, which then decides the best group of actuator nodes to perform the required task (see Figure 2.3(a)). In the distributed decision category, after receiving event information, actuator nodes communicate with each other and send sensed data, their residual energy, current position and action range.
2. LITERATURE REVIEW

Fig. 2.3: Actuator-actuator coordination mechanism: a) centralized, b) distributed.

to other actuator nodes in the network. This information is then used by actuators to determine whether to participate in an action (see Figure 2.3(b) [75]).

The advantage of centralized actuator-actuator coordination is that the decision center is able to select the best actuator nodes to carry out a task, especially when there are multiple events. This, however, results in an increased end-to-end delay as actuator nodes need to send their data to a central station. On the other hand, distributed actuator-actuator coordination reduces actuator response time in comparison to the centralized model because each actuator node is able to make local decision based on received data. On the other hand, the distributed model increases the energy consumption of actuator nodes as they need to communicate with each other after each event. Finally, there should be a mechanism in both coordination models to handle the occurrence of multiple events.

In summary, any developed techniques/protocols for WSANs must ensure reliable and real-time data communication between sensor and actuator nodes and also bounded by the task completion time. Moreover, they must address the following key issues [76][65]: 
2. LITERATURE REVIEW

- **Bounded Packet Delay.** Actuators have to act on sensed data quickly. Otherwise, it would be detrimental to the operation of a WSAN. Specifically, any communication protocols must ensure that packets do not exceed an application delay tolerance. For example, intelligent building applications require the control of water sprinklers within two seconds (see Table 1.1).

- **Reliability.** Sensed data and commands may be lost due to congestion, bit error, or bad connectivity. Therefore, protocols must be developed to address one or more of these issues such that sensed data and control information are communicated to sensor and actuator nodes reliably.

- **Bounded Task Completion.** Applications such as fire control systems require bounded task completion time. Specifically, they need a bound on the elapsed time from when sensor nodes report the occurrence of a fire to when water sprinklers completely extinguish the fire.

- **Network Lifetime.** This issue is similar to conventional WSNs, where the objective is to ensure that the remaining nodes run as long as possible. This problem is caused by difficulty of replacing the batteries of all sensor nodes, especially in high density WSANs.

- **Service Differentiation.** Multiple events with different urgency level should be treated accordingly. For instance, in home automation systems, there are sensors that are used to sense temperature and lighting, while some will be responsible for tracking the whereabouts of a person. These two types of sensor nodes generate traffic flows that require
different delivery time and task completion time.

Protocols developed for WSNs are generally not applicable to WSANs for a number of reasons. Sensors nodes are power constrained, and have limited transmission range and memory size, while actuator nodes are resource rich and have better transmission capabilities as well as buffer size. Hence, WSAN protocols must consider the resource constraints of both types of nodes. The information flow in WSNs is from sensor nodes to a sink, which form a many-to-one communication pattern. In a WSAN, the communication is many to many as it can take place between any nodes. Moreover, sensed data must be routed to the corresponding actuators that are able to mount the appropriate actions. Hence, localised, reliable and real-time communications are crucial to the operation of WSANs.

2.2.3 Sensor-Actuator Coordination

The fundamental problem addressed by past studies is determining energy efficient methods/protocols that allow sensor nodes to select and locate an appropriate actuator node. The main issues to consider include the coordination model used, i.e., single or multiple actuators, and how actuators update sensor nodes of their location.

Shah et al. [73] propose a cluster-based coordination and routing (CCR) protocol. Clustering is a standard approach used extensively in WSNs to decrease the energy consumption of sensor nodes. A number of clusters in a given network region are constructed based on a formula that considers the dimension of an area, the number of deployed sensor nodes, and transmission range of sensor nodes. Then for each cluster, a cluster head is selected based
on the residual energy of sensor nodes, data transmission rate and node
density. A new cluster head is selected once its residual energy becomes
the lowest in the cluster. The key roles played by cluster heads are data
aggregation, and forwarding packets on paths that satisfy end-to-end delay
that is within a given delay bound and least energy expenditure.

If there are multiple paths, a cluster head selects the path with nodes
having the most residual energy. The clustering method proposed in [73]
distributes energy consumption uniformly amongst nodes and prevents the
energy holes problem [78]. However, the most important issue is the over-
heads to forming a cluster. In [73], sensor nodes are required to collect
position information from all nodes in the network before they are able to
construct the optimal number of clusters for a given WSAN.

Melodia et al. [10] propose a distributed and adaptive event-based par-
titioning method for sensor-actuator coordination. When an event happens,
sensors in the event area independently determine the nearest actuator node.
More specifically, all sensor nodes in the event area that is sending data to
the actuator form a cluster and a delivery tree that is rooted at the actu-
ator node. The key advantage of this method is that sensors and actuator
within a given scope of the detected event are required to be active. This
means sensor nodes save energy when there are no events in the network
because there is no need to spend energy for cluster maintenance. On the
other hand, clusters are formed independently by sensor nodes using only
local information. Each sensor node selects the nearest actuator node as a
cluster head. This means sensor nodes are not required to communicate with
the other nodes to select cluster heads, as is the case in [73]. This method,
however, is not suitable for scenarios with frequent events as constructing clusters and locating the nearest actuator node will consume a significant amount of energy.

Zeng et al. [79] propose a real-time sensor-actuator coordination protocol. Sensor and actuator nodes are aware of their positions. In addition, actuator nodes are mobile. Actuator nodes periodically broadcast their position, residual energy and load. When sensor nodes receive the broadcast message, they select the nearest actuator node with the maximum residual energy and minimum load for event reporting. Interestingly, a sensor node is able to request an actuator to move closer whenever the end-to-end delay of its current path exceeds a given threshold. This model, however, is not energy efficient as it requires actuators to broadcast messages periodically, which creates significant overheads. Also, the movement of an actuator node in response to a sensor node request may lead to higher end-to-end delays.

Melodia et al. [26] propose a sensor-to-actuator framework that considers mobile actuator nodes, which comprises of a novel location management scheme, where an actuator broadcasts a message to inform all sensor nodes of its new position. Actuator nodes are aware of the location of all sensor nodes, and are equipped with two radios tuned to a distinct channel; one for communicating with sensor nodes, and other for communicating with other actuator nodes. To restrict the broadcast scope of an actuator’s messages to only relevant sensor nodes, the authors use Voronoi diagram to partition the network area into a number of convex polygons or scope. Each sensor node is then assigned to its nearest actuator node. Every time an actuator node changes its position, it communicates with other actuator nodes to deter-
mine the polygon of each actuator node. As a result, energy consumption is reduced because there is no need for sensor nodes to relay update messages.

Melodia et al. [26] also propose a method to reduce energy consumption when actuator nodes are mobile. Specifically, they use Kalman filter to predict the position of a mobile actuator at a given time. Actuators send update messages periodically, which is used by sensor nodes to predict an actuators future location. This has the effect of reducing location update messages, and as a result, there is less communication overhead. In the Voronoi diagram and Kalman filter method [26], each actuator node is aware of the position of all sensor nodes in the network. The location information is either setup manually, which is impractical, or communicated by actuators. The drawback, however, is that as the number of actuators increases, the network will incur a non-negligible amount of signaling overheads associated with localization.

Table 2.1 summarizes sensor-actuator coordination protocols according to their coordination, and actuator localization mechanism as well as whether actuators are mobile. All proposed models are based on multiple actuator coordination except for the CCR protocol [73]. As mentioned in Section 2.2.2, multiple actuator coordination incurs communication overhead, in addition to increasing the energy consumption of actuator nodes. The difference between static [73] and dynamic [10] clustering is whether sensor nodes communicate with each other to elect an actuator node as the cluster head. More specifically, in dynamic clustering, sensor nodes individually select an actuator node as a cluster head while in static clustering, all sensors in a cluster communicate with each other and choose an actuator node as their
2. LITERATURE REVIEW

Prior Works | Coordination | Actuator Localization | Actuator Mobility | Energy Conservation Techniques
--- | --- | --- | --- | ---
Melodia et al. [10] | Multiple Actuators | Dynamic Clustering | No | Rotate cluster head role and use alternate paths
Shah et al. [73] | Single Actuator | Static Clustering | No | Event-based Clustering
Zeng et al. [79] | Multiple Actuators | Actuator nodes periodically broadcast their position | Yes | No consideration
Melodia et al. [26] | Multiple Actuators | Voronoi diagram and Kalman filtering to predict position | Yes | Exploit Voronoi diagrams to scope broadcast messages

Tab. 2.1: Comparison of sensor-to-actuator proposals

destination. This means that dynamic clustering can be categorized as a multiple actuator coordination mechanism while static clustering is a single actuator coordination mechanism. Actuator localization, mobility and energy conservation are three key challenges in sensor-actuator coordination.

As shown in Table 2.1, all the proposed models, except Zeng et al. [79], provide an energy efficient way for sensor nodes to locate actuator nodes. However, the model proposed by Zeng et al. [79] is not energy efficient as it is only based on actuator mobility. The use of Voronoi diagram and Kalman filter, as proposed in [26] and [80], is promising for static WSANs. However, the effectiveness of these methods in WSANs with mobile actuators is not known, especially when the non-negligible amount of signaling overheads associated with localization is considered.

2.2.4 Routing Protocols

Routing protocols play a critical role in WSANs as sensed data or control messages have pre-determined Quality of Service (QoS) requirements. The main issues are route discovery, route maintenance, selecting a path that is within the required delay tolerance, and energy efficiency. Many routing protocols have been proposed for WSNs in recent years. However, according to
the literature none of the existing papers to date consider research challenges resulting from the coexistence of sensors and actuator nodes. That is, the communication path is ostensibly between sensors and one or more actuator nodes. For example, a sensor node can send data to multiple actuator nodes or select the best actuator node according to predefined parameters. Moreover, as actuator nodes are resource rich, they can bear the burden of the routing process. In addition, actuator nodes play the role of mobile sinks, which has a significant impact on how routing is carried out in a WSAN.

In the work reported in [81] the performance of three ad-hoc network routing protocols are analyzed in terms of their ability to decrease end-to-end packet delays. The study has its focus on analyzing Destination Sequenced Distance Vector (DSDV) [82], Dynamic Source Routing (DSR) [83] and Ad Hoc On-Demand Distance Vector (AODV) [84]. The authors use these protocols for actuator-to-actuator communication and coordination. DSDV is a proactive routing protocol that requires nodes to periodically exchange routing tables. DSR, a reactive routing protocol, embeds the route to be taken in the header of each packet. In addition, each node maintains a cache of routes that are compiled from the packets processed by them, and from those transmitted by the neighbouring nodes. AODV, another reactive routing protocol, works similar to DSR where route requests are sent in an on-demand manner. However, it neither maintains caches nor requires source routes in packets. The results produced in [85] show that at startup time, packets routed using DSR and AODV arrive quicker than those forwarded by DSDV. This is because routers using DSDV require a non-negligible amount of time to achieve route convergence. However, after the network becomes
stable, DSDV yields the lowest end-to-end delay as a source does not need to establish a route in an on-demand manner. The drawback of DSDV protocol is that it is not energy aware and wastes network bandwidth. To guarantee low end-to-end packet delays, sensor nodes are required to update their routing tables periodically, which leads to an increase in power and bandwidth consumption.

Hu et al. [80] propose the deployment of anycast and mobile actuator nodes to minimize energy consumption in WSANs. The proposed model is a variant of the Directed Diffusion (DD) [86] communication paradigm, which was proposed for WSNs. An anycast tree is built from a sensor to actuator nodes, where actuators form the leaves of the resulting tree. Every time an actuator node joins the network, it broadcasts a route discovery message. This causes all sensor nodes to update their anycast tree. In order to decrease energy consumption and communication overheads, an actuator node floods an explicit leave message to remove itself from the anycast tree. This model is suitable for applications that can tolerate minor packet loss. The main drawback is that actuator nodes may generate excessive signaling overhead when they change their location given the use of flooding to update their locations.

Boukerche et al. [87] propose a QoS aware routing protocol. Actuator nodes broadcast subscription messages with the energy level and hop count field set to zero. Each sensor node that receives this message updates these fields and collects information about its hop level from actuator nodes and the energy level of its neighboring nodes. Whenever a sensor node intends to send a data packet, it updates the packets delivery time based on the elapsed
time from when the packet is generated. Sensors uniformly distribute packets
to actuators based on packets delivery time in order to normalize their energy
usage. When paths have the same length, sensors will select the one with
the highest energy level. The QoS-aware source routing protocol selects a
path that meets a packets delivery time. There is no solution proposed
for the case when no path can be found. One mechanism to alleviate this
problem is for sensor nodes to increase their transmission power. However,
this needs a coordination mechanism that determines which nodes on a path
are required to increase their transmission power. Moreover, such mechanism
must balance energy consumption due to higher transmission power and end-
to-end delay.

Cayirci et al [88] propose a Power Aware Many-to-many Routing (PAMR)
protocol. The basic idea is to create a multicast tree rooted at a sensor node
towards actuator nodes using a publish and subscribe method. At start-up,
actuators broadcast their interest to sensor nodes. A sensor node with the
required data records the number of hops, minimum energy of nodes on the
route, and energy that will be consumed by nodes when forwarding packets.
A sensor node then selects the route with the smallest weight to minimize
the formation of energy holes, and also provides end-to-end packet delay. In
other words, each sensor node builds an energy efficient delay aware multicast
tree to actuator nodes that are interested in its sensed data. The drawback of
this model is that PAMR cannot guarantee end-to-end packet delays. This
is because sensor nodes have a fixed transmission power, and they do not
have any option when the delay on the minimum weight path is larger than
required packet delivery time.
To overcome this problem, the authors also propose a different version of the algorithm called PAMR protocol Power Controlled PAMR (PCPAMR), where sensor nodes increase their transmission power when the required delivery time of a packet is less than the delay of an existing path. In other words, PCPAMR balances the required end-to-end packet delay and energy consumption of sensor nodes. The problem with PCPAMR is that when each sensor node individually changes its transmission power, it causes non-uniform energy consumption and energy holes to occur along the packet forwarding path.

In [89] a Delay-Energy Aware Routing Protocol (DEAP) is proposed. Sensor nodes wake up once in every predetermined time period, transmit data and then go to sleep. The time length of an active period depends on the number of data packets in the transmission queue. A large queue size causes a long active time. Hence, sensor nodes make local decisions whether to sleep or to be active based on their queue length. Whenever a sensor node intends to send data packets, it selects one that is closest to an actuator node.

DEAP distributes energy consumption between a source nodes forwarding set because each time a source node wants to send data packet, its forwarding set may change based on the active and sleep period of its neighbours. DEAP prolongs WSAN lifetime and reduces energy consumption in sensor nodes but at the expense of end-to-end packet delay. This happens in normal scenario due to the action of source node building the shortest forwarding path to the nearest actuator node. However, when sensor nodes go to sleep according to their load, other nodes have to rebuild their routing path as nodes may
decide to enter sleep mode, or new nodes may be available.

In applications such as the water irrigation system [21], actuators are in charge of opening or closing valves depending on the need to water areas as determined by sensed data provided by sensors. This implies that sensors may send data to multiple actuators. Therefore, in order to save energy, Sanchez et al. [90] propose an energy-efficient multicast routing protocol. As the problem of finding an energy-efficient multicast tree is NP-complete, a heuristic method that incorporates nodes locations to build energy-efficient multicast paths is deployed. The heuristic uses the destination and energy cost to each destination during tree construction and merging all the paths into a common node, which then serves multiple destinations. The Multicast problem proposed by Sanchez et al., however, is not delay-aware. When the number of actuator nodes increases, the computational power and storage requirement associated with finding candidate sets and conducting a merge become significant for sensor nodes.

Table 2.2 summarizes the aforementioned routing models according to their route discovery and maintenance mechanism. The table also compares the approach used to ensure that the end-to-end packet delay of a path is bounded. A notable approach in this respect involves actuator nodes issuing feedback to sensor nodes on a given path to increase their transmission power until the required end-to-end delay path is met. On the other hand, protocols that place the responsibility of route discovery and maintenance on sensor nodes have high communication overheads and energy consumption.
Prior Works | Route Discovery | Route Maintenance | Delay Bound | Energy Conservation Techniques
---|---|---|---|---
Dinh et al. [81] | Sensor nodes | Sensor nodes periodically exchange routing tables | No | None
Hu et al. [80] | Actuator nodes | Actuator nodes | Select the nearest actuator node | Anycast tree
Boukerche et al. [87] | Actuator nodes | Actuator nodes periodically broadcast subscription messages | Select the path that meets packet delivery time | Distributes packets between existing paths
Cayirci et al. [88] | Actuator nodes | Actuator nodes periodically broadcast subscription messages | Vary transmission range | Multicast tree
Durresi et al. [89] | Sensor nodes | On-Demand | No | Sleep and wake-up
Sanchez et al. [90] | Sensor nodes | None | No | Multicast tree

Tab. 2.2: Comparison of routing protocols

### 2.2.5 Transport Protocols

Reliability is a critical issue as actuators use sensed data to reconstruct events before launching the corresponding action(s) for a given event. Therefore, it is important that actuators ascertain the type, location and intensity of events reliably. The main approaches applied in earlier studies are based on either varying the transmission power of nodes, or the number of retransmissions. Apart from that, service differentiation is a key concern in scenarios with multiple events. In general, transport protocols should provide both reliability and real-time data communication between sensor and actuator nodes. In addition, the protocols must also guarantee event reliability; defined as the amount of data that is reliably delivered to an actuator node within a given time bound. In addition, anycast communication is used frequently as sensed data should be routed to the closest actuator node(s). For these reasons, transport protocols developed for WSNs are generally not suitable for use in WSANs.

Zhou et al. [91] propose a real-time data transport protocol called POWERSPEED. Sensor nodes calculate the hop-by-hop delay that a packet will
experience on a given path. The expiration time of each packet is then calculated to determine the transmission power of the nodes. A packet that has a higher delay tolerance traverses more hops as sensor nodes will use a lower transmission power, which increases the number of hops on a given path. Otherwise, the packet will be forwarded over a route with fewer hops. The limitation of POWER-SEED is that it does not consider congestion. In other words, increasing or decreasing transmission power does not necessarily alleviate or avoid congestion on a given route. In addition, POWER-SPEED cannot handle multiple events.

Gungor et al. [92] propose RT2, an energy efficient, reliable and real-time transport protocol. The authors define event reliability as the percentage of event data received by actuator nodes within a given time bound. To provide event reliability, RT2 uses the Time-Critical Event First (TCEF) [92] scheduling algorithm, where sensor nodes service packets according to their deadline. An interesting feature is that actuator nodes send feedback messages to sensor nodes in order to decrease their sampling or transmission rate if the observed event reliability is above a given percentage. Sensor nodes are notified of impending congestion when their buffer overflows or when the average packet delay exceeds a threshold. In this situation, sensor nodes set a Congestion Notification (CN) bit in their packets to inform actuators of the impending congestion [93]. Upon receiving packets with the CN bit set, and observing that the event reliability is below a given threshold, actuators inform sensor nodes to decrease their sampling or transmission rate. The advantage of RT2 is that it saves the energy of sensor nodes when there is congestion. On the other hand, its drawback is that there is no mechanism
to overcome congestion and guarantee event reliability.

Ngai et al. [94][95] propose a latency-oriented fault tolerant (LOFT) transport protocol for WSANs. Sensors and actuators are aware of their locations. Each sensor node maintains and schedules queues according to the urgency of events. When a sensor node intends to send a data packet to a destination, it selects the path that meets the required delivery time. In scenarios where there are multiple paths, the source node uses the least congested path. It is also proposed to use mobile actuators to service areas with high frequency of events. The drawback of [94][95] is that sensor nodes are required to send the status of their queue to their neighbors, which increases signaling overheads. It is also assumed that nodes are aware of their locations. For example, the use of the Geographic Positioning System (GPS), which consumes non-negligible amount of energy is suggested.

Ngai et al. [94][95] also propose a novel replication method to increase reliability. Nodes maintain the link loss rate to each of their respective neighbor, and use this information to determine their transmission strategy. If the loss rate is high, a sensor node sends its packets to multiple neighbors that have a path to the actuator. If all neighbors fail to meet the required loss rate, the sensor node sends a feedback message to a packets previous hop. The process is repeated until another path is found or the feedback message is received by the source node, which then decides whether to terminate the transmission.

Melodia et al. [10] propose a transport protocol that provides event reliability in applications. Here, event reliability is defined as the percentage of data that arrives within a given time bound. Whenever a sensor node needs
to transmit a data packet, it builds a routing path to its nearest actuator using the minimum transmission power. When an actuator receives a data packet from a sensor node, it computes its event reliability and broadcasts the result to sensor nodes. If the event reliability of an application is not met, sensor nodes adjust their transmission power according to a given probability.

For example, if the calculated event reliability is 0.1 of required event reliability, more sensor nodes in the packet forwarding path increase their transmission power. On the other hand, if the calculated event reliability is 0.9, sensors reduce their transmission power. This mechanism, however, does not consider congestion at nodes. Hence, increasing transmission power may not necessarily alleviate the low event reliability reported by actuators. In addition, the authors assume that in each network area, only one event will occur, and do not consider the possibility of multiple events requiring different reliability bounds within an area. For example, in some applications such as home automation, light or vision sensors will generate different sensed data according to different event reliability requirements.

Melodia et al. [26] propose a transport protocol for sensor-actuator network similar to the distributed heuristic model in [10], which trade-offs the energy consumption of sensor nodes and provides minimum event reliability. The only difference is that whenever the reliability is low, even after an increase in transmission power, the actuator node detects congestion occurrence. In this scenario, a new actuator is chosen, and half of the traffic is routed to the selected actuator. The limitation with their approach is that the new actuator is chosen with respect to the old actuator. Hence, the new actuator may not be close to sensor nodes. As a result, the redirected traffic
may unnecessarily consume more energy.

Table 2.3 summarizes transport protocols developed for use in WSANs. Most of them rely on the existence of multiple paths, and the use of varying transmission range to shorten forwarding paths. The former technique means more nodes are required to participate in the forwarding of packets, whereas the latter method leads to increased interference. Moreover, the use of multiple paths and increased transmission range is likely to degrade network capacity as nodes on different paths are likely to interfere and increase contention delay. Apart from that, transport protocols that rely on retransmissions may cause unwanted responses. For example, in a fire system, a temperature sensor node may send a data packet on two different paths, and as a result, they have different arrival times. Upon receiving the first packet, the actuator starts the water sprinklers. However, when the second packet arrives, it may conclude that more water is required to extinguish the fire. It thus increases the water flow to the sprinklers, which unfortunately leads to flooding.

2.2.6 Actuator-to-Actuator Coordination

In actuator-to-actuator coordination, it is critical that tasks are completed on time. Hence, bounding this time is one of the most important application requirements to be met. Vassis et al. [96] propose a multi-channel actuator-actuator communication protocol to decrease task completion time. Each actuator uses two independent radio transceivers, with one used for communicating with single-hop or neighboring actuators, and the other for multi-hop or remote actuators. The transceiver used for multi-hop commu-
Tab. 2.3: Comparison of transport protocols

<table>
<thead>
<tr>
<th>Prior Works</th>
<th>Delay Bound</th>
<th>Reliability</th>
<th>Service Priority?</th>
<th>Energy Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhou et al. [91]</td>
<td>By changing transmission range of sensor nodes to shorten or elongate a forwarding path</td>
<td>No consideration</td>
<td>Yes</td>
<td>Forwarding paths are selected based on required delivery time of packets</td>
</tr>
<tr>
<td>Gungor et al. [92]</td>
<td>Scheduling packets according to their deadlines and dynamic adjustment of sampling rate</td>
<td>TCEF scheduling algorithm and the changing sampling rate</td>
<td>Yes</td>
<td>During congestion, sampling rate of sensor nodes is reduced</td>
</tr>
<tr>
<td>Ngai et al. [94][95]</td>
<td>Forwards packets on the least congested path. They also consider the use of mobile actuators</td>
<td>Measuring links loss rate and sending multiple copies of a packet along many paths</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Melodia et al. [10]</td>
<td>Probabilistic transmission schedule, and increasing nodes’ transmission range to reduce route length</td>
<td>Probabilistic recruitment of sensor nodes</td>
<td>No</td>
<td>Adjust sensor nodes’ transmission range</td>
</tr>
<tr>
<td>Melodia et al. [26]</td>
<td>Changing sensor nodes’ transmission range and shifting load to less congested actuator nodes</td>
<td>Changing sensor nodes’ transmission range and shifting packets onto less congested paths or actuators</td>
<td>No</td>
<td>Reduce sensor nodes’ transmission range if event reliability is above a given threshold</td>
</tr>
</tbody>
</table>

Prior to these new protocols, the network was congested due to separate collision domain used for local and remote communications. As a result, nodes experience shorter medium access delay. On the other hand, the multihop transceiver with its longer transmission range reduces end-to-end delay as packets traverse fewer hops to their destination actuator. The downside of this approach is that it incurs additional cost and energy due to the use of two radio transceivers.

The authors of [10] propose a localized auction algorithm to minimize task completion time. Each actuator node is assigned to a given area. As actuators may have overlapping areas, the algorithm selects one that can complete the task with minimum energy expenditure and within a given delay bound. The selection is carried out according to actuator nodes residual energy and minimum load. This approach can only be used when there are multiple static actuators in an overlapping area. The proposed algorithm
is not capable of handling the occurrence of multiple events with different
task completion times because it does not prioritize tasks according to their
urgencies.

Zeng et al. [79] propose an approach that uses mobile actuator nodes. The first actuator node that receives an event occurrence report starts collecting information such as residual energy and current position from other actuators. Actuators also indicate whether they have received the same report. Based on the collected information, an actuator that has the highest residual energy and located in range of the event area is then assigned to complete the task. A key limitation of this work is that the authors have not considered multiple events. In particular, an actuator may be selected to carry out multiple tasks but its residual energy may only be sufficient to complete one task.

Melodia et al. [26] propose a novel mechanism to control mobile actuators in scenarios where events occur in partially overlapped space and/or time. They formulate a Mixed Integer Non-Linear Program (MINLP) that aims to minimize energy consumption whilst ensuring task completion within a given time bound. In the presence of multiple events, the proposed mechanism aims to guarantee task completion time for high priority events. For each task, the proposed model selects an ideal number of actuators and determines the actuator velocity required to move into an event area using minimal energy whilst adhering to a given delay bound. The key limitation of this work is that the MINLP model is a centralized method, and the authors do not propose any heuristic or distributed models for the problem.

Ngai et al. [94] develop a relocation method where actuators are mo-
bile and visit areas that have a high frequency of events. The network area is divided into cells, and a selected actuator periodically collects the event frequency of all cells in the network. The selected actuator node runs a re-location algorithm to balance load and move more actuators to areas with higher event frequencies. This method requires actuator nodes to periodically communicate with selected actuator nodes, which increases energy consumption. To reduce power consumption, the authors assume fixed event frequency. This means that after an actuator node is assigned to a given area, there is no longer any need for communication between actuator nodes. Unfortunately, in some applications, different areas will have varying number of events occurring within a given time period.

Shah et al. [73] also suggest an actuator relocation method. The network area is divided into clusters, where cluster heads are responsible for collecting sensed data from their respective cluster, and for sending them to actuator nodes. An actuator-actuator coordinating procedure is triggered when a cluster head is not in the action range of any actuator nodes. This cluster head then causes a relocation message to be broadcast to other actuator nodes, which then determine the possibility of covering the cluster head while remaining in a range to cover the existing cluster heads. If an actuator node can move toward the cluster head that issues the relocation message, it sends back a reply message that contains its residual energy and number of clusters it is responsible for. The actuator node with the maximum amount of residual energy and minimum number of attending cluster head(s) is then selected to help the cluster head. The disadvantage of this actuator relocation model is that while actuator nodes are mobile, the authors do not propose any solution
when there are no actuators that could move toward uncovered cluster heads.

Table 2.4 summaries actuator-to-actuator coordination models. As shown, only a handful of works address the research issues rasied in this area. All the studies have the common theme of developing protocols to control mobile actuator nodes. A key observation is that guaranteeing task completion time when there are multiple events is currently not possible with distributed actuator-to-actuator coordination mechanisms. Besides, coordinating multiple actuators, especially in scenarios with varying event frequencies, incurs a significant amount of communication overheads. On the other hand, centralized coordination models have high delays and as a consequence, they are not suitable for tasks that require fast completion time.

### 2.2.7 Summary

The potential of WSANs has attracted the attention of researchers from diverse disciplines. The key driving factor is their wide range of applications as sensor nodes are now able to interact with the environment they are in by affecting the key parameters. An example is the use of water sprinklers to reduce temperature. In this context, this section has provided an extensive review of solutions pertaining to the coordination process between sensor
and actuators, as well as between actuators. It is evident that addressing the coordination problem will have to involve energy efficient protocols from a number of protocol layers. Moreover, in comparison to WSNs, a significant amount of research remains. In particular, those related to developing distributed protocols to control multiple actuator nodes, and providing an upper bound on task completion time.

2.3 Mobile Sink

2.3.1 Introduction

A number of previous studies, such as [33] [97] [98] [50], have investigated the use of one or more mobile sinks to survey and collect sensed data directly from sensor nodes. Figure 2.4 shows the feasible sites of a mobile sink in a typical WSN. In this diagram, the squares denote the feasible sites that the mobile sink will visit and stop for data collection. The data forwarding path from sensor nodes to the sink is dependent on the current position of the sink. This requires sensor nodes to dynamically plan one or more data forwarding paths to each feasible site whenever the sink node changes its position over time. As demonstrated by [33], a mobile sink that moves at the periphery of a sensor field maximizes the lifetime of sensor nodes. Intuitively, by changing the position of the sink over time, the forwarding tree will involve a different set of sensor nodes, and hence, will help to balance energy consumption [51] [52] [53] [99] [100].

The travel path of a mobile sink depends on the real-time requirement of data produced by nodes. For example, in hard real-time applications such
as a fire detection system [101], environment data needs to be collected by a mobile sink quickly. Moreover, a mobile sink node may change its position after a certain period of time and select another data collection/feasible site. The feasible sites and corresponding sojourn time are dependent on the residual energy of sensor nodes [66] [54] [55] [56] [57]. In general, limitations such as the maximum number of feasible sites [102], maximum distance between feasible sites and minimum sojourn time [54] govern the movement of a mobile sink.

In delay tolerant applications such as precision agriculture [20][5], sensed data can be delayed before delivery to the sink node. In order to use a mobile sink for such applications, a fundamental problem is to determine how the mobile sink collects sensed data. One approach is to visit each sensor node to receive sensed data directly [37]. This is essentially the well known Traveling Salesman Problem (TSP) [103], where the goal is to find the shortest tour that visits all sensor nodes. However, with increasing number of nodes, this
problem becomes intractable and impractical as the resulting tour length is likely to violate the delay bound of applications. To this end, researchers have proposed the use of Rendezvous Points (RPs) to bound the tour length [36] [34]. This means a subset of sensor nodes are designated as RPs, and non-RP nodes simply forward their data to RPs. A tour is then computed for the set of RPs as shown in Figure 2.5. As a result, the problem, called rendezvous design, becomes selecting the most suitable RPs that minimize energy consumption in multi-hop communications whilst meeting a given packet delivery bound. A secondary problem here is to select the set of RPs that result in uniform energy expenditure amongst sensor nodes in order to maximize network lifetime.

Existing methods that employ a mobile sink can be grouped into two categories: (i) Real-time applications, where a mobile sink moves between the feasible sites and settles in each site to collect sensed data, and (ii) Delay-
tolerant applications, where a mobile sink visits sensor nodes in network area and collects sensed data. The main goal of protocols in category (i) is to mitigate the energy holes problem by changing the position of the sink node. In particular, they move the mobile sink to an area where nodes have large residual energy. Protocols in category (ii) aim to find a travel path for a mobile sink that minimizes energy consumption whilst adhering to the delay bound provided by an application [51]. In the following sections, challenges faced by these protocols are reviewed.

2.3.2 Real-Time Applications

The first mobile sink approach for real-time applications is proposed by Gandham et al. [104]. In this study multiple mobile sinks are deployed that periodically move between feasible sites, spread over a given area. The Base Station Location (BSL) problem is formulated as an Integer Linear Program (ILP) in order to find the sojourn time of mobile sinks for each feasible site. The objective of the ILP is to minimize sensor nodes energy consumption rate. The input to the ILP is the position of sensor nodes and their current residual energy. However, the solution to the ILP does not necessarily mitigate the energy holes problem. In [66], an alternative approach is introduced in which the problem is formulated as a Linear Program (LP) where the objective is to maximize network lifetime. In this formulation, each sensor node is a potential feasible site. An LP is then used to calculate the maximum sojourn time of the mobile sink at each sensor nodes position.

The main problem with [66] and [104] is that they do not take into consideration the travel time of the sink between two consecutive feasible sites.
In large WSNs, this travel time may violate applications delay constraint. In addition, from an energy consumption point of view, the cost of rebuilding routing paths whenever a mobile sink moves between sites is not considered. If a mobile sink has a short sojourn time, the amount of energy spent rebuilding new routing paths may offset the energy saved from having a mobile sink shorten multi-hop communications.

These problems are overcome in [54] using a localized distributed protocol for sink mobility. The problem is addressed as a Mixed Integer Linear Programming (MILP) with the objective of prolonging network lifetime. The concept is implemented in a WSN deployed over a grid. In the proposed strategy, the next destination of the mobile sink is limited to neighboring sites. A mobile sink is also required to stay at a site for a minimum period to amortize route reconstruction cost and to conserve its energy. The outputs from the MILP include a set of selected sites to be visited in order by the mobile sink as well as the corresponding sojourn time. As the MILP cannot be solved efficiently, the authors of [54] propose a Greedy Maximum Residual Energy (GMRE) heuristic algorithm. For each feasible site, there is a dedicated sensor node that gathers residual energy from surrounding sensor nodes. As mentioned in Chapter 1, one hop sensor nodes around the sink have the highest energy consumption rate as compared to other nodes. GMRE checks the residual energy of neighbor sites periodically and moves the mobile sink to the site with a higher residual energy than the current site.

Half-quadrant-based moving strategy (HUMS) for mobile sink is proposed in [55]. HUMS moves a mobile sink towards sensor nodes with the highest
residual energy and avoids passing through nodes with the lowest residual energy. Sensor nodes send their residual energy and position information to the mobile sink. Therefore, a mobile sink knows the position of the sensor node with the highest residual energy, and is marked as ‘destination’. In addition, the mobile sink knows the sensor nodes with the lowest residual energy in each data forwarding path, which are termed as quasi-hotspots. The mobile sink moves one hop toward the destination node in each movement. In order to move toward destination nodes and avoid crossing quasi-hotspots, HUMS divides the network area into eight half-quadrants and approaches the destination node through the half-quadrants with the highest residual energy.

Liang et al. [105] bound the total distance travelled by a mobile sink. Specifically, the mobile sink is powered by either petrol or electricity, both of which are finite resources. The authors have formulated the distance-constrained mobile sink problem as a MILP. As the problem NP-hard, they propose a three-phase heuristic algorithm as an alternative solution. In the first phase, a heuristic algorithm calculates the maximum mobile sink sojourn time for each sensor node. In the second phase, the algorithm finds a feasible travel path for the mobile sink to stop in a subset of positions such that the total mobile sink sojourn time is maximized. In the third phase, the algorithm calculates the exact sojourn time at each chosen sensor node’s position.

Multiple mobile sinks scenario has been considered by Marta et al. [106]. Each mobile sink broadcasts a join message as a cluster head to make its own cluster. Among the join messages received from mobile sinks, sensor nodes
select the message from the cluster with the closest cluster head. Each mobile sink moves into another part of the network and creates new cluster when the energy level of its one-hop neighbor sensor nodes drops below a certain level. Mobile sinks are equipped with two wireless transmitters, one short range for communicating with sensor nodes and one long range to communicate with other mobile sinks.

Table 2.5 compares proposed mobile sink moving patterns in real time applications. Apart from the first two algorithms proposed in [104] and [66], all the methods consider the travel time between two consecutive sites and also minimum sojourn time at each selected site. The method proposed by Liang et al. [105] is the only algorithm that has no bound on the total travel time of the mobile sink. Gandham et al. [104] and Marta et al. [106] adopt multiple mobile sinks to change the data forwarding pattern into many-to-many, which helps balance energy usage and increase network lifetime.
2.3.3 Delay-Tolerant Applications

Existing mobile sink methods for delay-tolerant applications can be grouped into two categories: (i) direct, where a mobile sink visits each sensor node and collects data via single-hop, and (ii) rendezvous, where a mobile sink only visits nodes designated as RPs. The main goal of the protocols in category (i) is to minimize data collection delay, whereas those in category (ii) aim to find a subset of RPs that minimize energy consumption whilst adhering to the delay bound provided by an application [51]. The following sections review these protocols in detail.

2.3.3.1 Direct

Initial studies expect a mobile sink to visit sensor nodes randomly and transport collected data back to a fixed sink node. An example is the use of animals as mobile sink nodes to assist in data collection from sensor nodes scattered on a large farm [50]. To reduce the latency of visiting each sensor node, researchers have proposed Traveling Salesman Problem (TSP)-based data collection methods. In essence, the problem is reduced to finding the shortest travel path that visits each sensor node [37]. For example, Traveling Salesman Problem with Neighborhood (TSPN) [107] involves finding the shortest travel tour for a mobile sink node that passes through the communication range of all sensor nodes. Another TSP-based algorithm [108], called label-covering, considers a WSN as a complete graph. For each edge, it calculates a cost and associates a label set. An edge is a single hop data forwarding path between two sensor nodes which means both sensor nodes
are in communication range of each other. The cost of an edge is the Euclidean distance between nodes, whereas the label set contains sensor nodes whose transmission range intersects with the given edge. The label-covering algorithm selects the minimum number of edges where their associated label set covers all sensor nodes.

In some applications such as green-house management system [22], the sampling rate of sensor nodes may not be uniform. This is because sensor nodes in different parts of the network may have a different sampling and data generation rate. Consequently, visiting sensor nodes with the same frequency rate by a mobile sink node may cause buffer overflow. To this end, the authors of [85] propose a visiting schedule plan for a mobile sink to visit some sensor nodes more frequently than others such that none of the sensor node’s buffer overflows. The proposed scheduling algorithm is based on Earlier Deadline First (EDF) [109]. Sensor nodes are divided into subsets such that the sampling rate of those in one subset is the same. However, the authors have not considered the mobile sink’s speed limitation. In practice, the long travel time between two consequence partitions may cause sensor nodes’ buffer to overflow.

In a different work, Gu et al. [110] propose a partitioning-based scheduling (PBS) algorithm that considers the location of each partition. The authors partition sensor nodes into several groups with respect to their data generation rate and location. The scheduling algorithm schedules the mobile sink’s visiting sequence in a manner that minimizes the overhead of moving back and forth across far-away nodes. PBS concatenates all groups and forms a mobile sink path so that all nodes can be visited frequently to prevent
buffer overflows.

2.3.3.2 Rendezvous

The problem with collecting data directly from sensor nodes is that it becomes impractical when there are a large number of sensor nodes. Visiting each sensor node increases the mobile sink’s travel path length and also results in sensor nodes experiencing buffer overflow due to data collection delays. In order to address this problem, researchers have proposed a rendezvous based model, in which a mobile sink only visits a subset of sensor nodes called RPs. The sensor nodes outside the mobile sink path send their data via multi-hop communication to these RPs. Studies such as, [36] [34] [35] [58] [111] [59], deploying this approach can be classified according to the mobile sink’s trajectory; that is, whether it moves along a fixed path, or its path is unconstrained by any external factors.

2.3.3.3 Fixed

In the studies conducted in [58] [111] and [59], a mobile sink has a fixed path, and sensor nodes are deployed randomly in the vicinity of its travel path. Sensor nodes that are inside a mobile sink’s communication range play the role of RPs, and collect data from other sensor nodes. An example application is a traffic management system where mobile sinks are public buses that roam a city to collect data from sensor nodes placed on buildings [58]. In these approaches, the length of the travel path is not dependent on the buffer size of sensor nodes or application deadline. Hence, the buffer of RPs may overflow or packets may expire before data is collected by the sink.
Xing et al. [35] propose Rendezvous Design for Fixed Tracks (RD-FT), where the movement of a mobile sink is governed by the application deadline. They also consider obstacles that restrict the movement of a mobile sink along a pre-defined path. The objective is to find a set of RPs on the fixed path such that the length of data forwarding paths from sensor nodes to RPs is minimized and the travel time between RPs is limited to the required packet delivery time.

### Unconstrained

In [35], a WSN with a static sink node and a Mobile Element (ME) is assumed to collect data from RPs. Moreover, RPs perform data aggregation. An algorithm called Rendezvous Design for Variable Tracks (RD-VT) is proposed with the objective of identifying a travel path that is shorter in duration than the packet delivery time. The algorithm first constructs a Steiner Minimum Tree (SMT) rooted at the sink node. RD-VT then starts from the sink’s position and traverses the SMT in pre-order until the shortest distance between visited nodes is equal to the required packet delivery time. Since in a SMT, a Steiner point may be a physical position and does not correspond to the position of a sensor node, RD-VT replaces these virtual RPs with the closest sensor nodes. A major limitation of RD-VT is that traversing the SMT in pre-order leads to the selection of RPs that in turn results in long data forwarding paths to sensor nodes located in different parts of the SMT. As a result, RD-VT causes nodes to have an un-balanced data forwarding load and energy consumption.

Xing et al. [36] propose Rendezvous Planning with Constrained Mobile
Element Path (RP-CP). Similar to RD-VT, the authors consider a WSN with a fixed sink node and a Mobile Element (ME). The RP-CP first constructs a routing tree that is rooted at the sink node and connects all sensor nodes. Then, each edge of the routing tree is assigned a weight that corresponds to the number of nodes that use that edge to forward their data to the sink node. The ME is restricted to moving only on the edges of the tree. To construct the ME’s travel path, RP-CP first sorts all edges according to their weight. It then selects the edges with the highest weight until the length of the selected edges becomes equal or less than the required packet delivery time. The problem with RP-CP is that the travel path of the ME is restricted to routing tree edges. This also means that the ME will visit the sensor nodes on the selected edges twice.

In [36], the authors propose an improvement to RP-CP, called Utility-based Greedy (RP-UG). Initially, a geometric tree, which is rooted at the fixed sink node, is constructed and all edges on the tree are split into multiple, short intervals, denoted as $L_a$. All points that join two edges with length $L_a$ are considered as RP candidates. RP-UG starts from the sink’s position and in each step, adds a RP that has minimal distance to sensor nodes and also results in the shortest travel tour between RPs. RP-UG uses a TSP solver to calculate the tour length. To finalize the tour, RP-UG replaces virtual RPs with the closest sensor nodes and marks them as RPs. RP-UG does not balance the energy consumption rate of sensor nodes, which has a significant impact on network lifetime. Specifically, the network lifetime is determined by the sensor node with the highest energy consumption rate, say $n$, assuming that all nodes have the same initial energy level. In this regard,
2. LITERATURE REVIEW

RP-UG does not aim to minimize the energy consumption rate of node \( n \). In addition, when using RP-UG with a small \( L_o \) value, the number of RPs increases significantly and the complexity of RP-UG grows exponentially. The algorithm uses a TSP solver \( N \) times in each iteration, where \( N \) is the number of RPs. Hence, RP-UG has a running time complexity of \( O(N^2 \times O(TSP)) \).

A Cluster-Based (CB) algorithm is proposed in [34], whereby a binary search procedure is used to select RPs. Figure 2.6 shows how CB works in a network with 10 nodes where the maximum allowed tour length is 90m. In the first iteration, based on a binary search on the range from 0 to 10, CB selects five random cluster heads. In this case, cluster heads are node 2, 3, 4, 5 and 7 (Figure. 2.6(a)). Other nodes then establish a path to their closest cluster head in terms of hop-count. After the clusters are determined, CB starts from the sink node’s position and selects one node from each cluster as an RP, which is the closest node to the set of selected RPs. Figure. 2.6(b) shows that node 8 from cluster \{7, 8\} which is the closest node to the sink is selected as an RP and also node 6 from cluster \{5, 6\} and so on. This diagram also indicates that the final tour has a length of 127m, which is larger than 90m. Therefore, CB reduces the number of clusters to two. According to Figure. 2.6(c), node 7 and 1 are selected randomly as new cluster heads. Figure. 2.6(d) shows that the shortest possible tour between clusters is a tour including nodes 2 and 9 that has a length of 55m. CB therefore increases the number of clusters to three because the tour length is less than 90m. Figure. 2.6(e) shows 4, 3 and 10 are selected as new cluster heads and the final tour length is 128m which is larger than 90m. This causes CB to reduce the
number of clusters to two again. However, given that CB has already found a tour for two clusters, it stops and outputs \{2, 9\} as the final tour. Note that this tour does not pass through dense parts of the network consisting of nodes with larger number of neighbors such as nodes 10, 6 and 7. This problem causes long data forwarding paths from sensor nodes to the RPs and non-uniform energy depletion, which reduces the lifetime of the WSN.

### 2.3.4 Discussion

In delay-tolerant applications, visiting sensor nodes directly, fully mitigates energy holes, as shown in [50] [37] [107] [108]. However, this approach is not practical when there are a large number of sensor nodes, meaning the mobile sink is unlikely to meet the required data delivery time. Therefore,
in order to be able to alleviate energy holes as well as meet delivery time, researchers have proposed a combination of single-hop and multi-hop data forwarding patterns. Programming a mobile sink to visit dense parts of a WSN is critical because sensor nodes generate/forward the highest number of packets. Hence, giving priority to sensor nodes in these parts during tour computation will help to reduce congestion points, and in turn to reduce energy consumption and to improve WSN lifetime. In addition, it helps to mitigate the energy holes problem. As it is shown in Chapter 3, this observation is exploited to produce a hybrid, unconstrained movement pattern for a mobile sink that visits nodes that forward high number of data packets. RP-UG [36] minimizes network energy consumption by reducing the physical distance between sensor nodes and RPs. However, due to the existence of obstacles, physical distance is not a reliable indicator of energy consumption.

2.4 Mobile Charger

2.4.1 Introduction

Battery powered sensor nodes have finite lifetime. To overcome this problem, researchers have considered energy harvesting methods such as solar [47] [48] or wind [49]. A $37\,mm \times 33\,mm$ solar cell can produce an average 655 mWh of energy on a sunny day, which is sufficient to support a TelosB wireless modules send and receive operation for 9.2 hours [47]. However, such energy can be harvested during the day provided that the node is exposed to the sun. Another solution is to replace the battery of sensor nodes, which is made difficult or impractical given the large number of sensor nodes and possible
remoteness of their operating environment.

To date, there are three wireless charging technologies for charging the battery of sensor nodes: inductive coupling, electromagnetic radiation and magnetic resonant coupling. In inductive coupling [112] [113] [114], the primary coil in the transmitter generates a high frequency electro-magnetic field that induces a voltage in a secondary coil at the receiver. The distance between the two coils must not exceed 0.16 mm. To date, inductive coupling is used successfully in a number of applications such as electric toothbrush [115] and medical implants [116]. However, it is not applicable to charging wireless sensor nodes due to its distance requirement.

Electromagnetic radiation [117] refers to the transfer of energy through electromagnetic waves. The main problem with the electromagnetic radiation is its low efficiency. He et al. [118] record an energy transfer efficiency of only 1.5% when the distance between source and receiver is 30cm.

A recently developed wireless charging technology is magnetic resonant coupling [119]. This system consists of a magnetic resonant coil at the source and one at the receiver working at the same frequency. Kurs et al. [119] show an efficiency of 40% when the distance between the source and destination is less than two meters, which is much higher than electromagnetic radiation. In addition, magnetic resonant coupling does not need strict alignment between the source and receiver. In other words, a source node does not need to know the exact position of a receiver, and thereby, making it more applicable to WSNs as sensor nodes are scattered in an environment randomly.

Figure 2.7 shows an example of a mobile wireless charger used to replenish the batteries of sensor nodes. In addition to sensing and transmission of
collected data, sensor nodes inform the sink node of their residual energy and energy consumption rate. The sink node then runs an algorithm to determine the best tour that allows the mobile charger to replenish the batteries of sensor nodes. The mobile charger returns to the depot (or sink’s location) after completing its tour to recharge its own battery.

Despite its potential and advantages, magnetic resonant coupling cannot be deployed effectively unless a number of outstanding issues are addressed:

- The set of sensor nodes to be visited and the charging time of each sensor node should be determined. The ideal solution is to charge all sensor nodes up to their maximum battery capacity [62]. This, however, may not be achievable if there is a large number of sensor nodes or when the mobile charger has finite battery capacity. Hence, the mobile charger is constrained to charge only a subset of sensor nodes. The problem then becomes finding the best subset of sensor nodes that effectively utilize the energy provided by the mobile charger.

- In order to conserve energy, the distance travelled by the mobile charger
should be minimized. A mobile charger expends energy when it travels from one node to another. Therefore, a key issue is to minimize energy consumption due to travel and consequently maximize the available energy of charged sensor nodes.

To this end, the studies conducted in [60] [61] [62] [63] [120] [121] [122] [123] aim to address the aforementioned issues to some extent. A set of sensor nodes are visited by one or more mobile chargers and their batteries are charged to a certain level. Their charging algorithms assume the mobile charger has a limited or unlimited battery capacity. In addition, a method is developed to charge only a subset of nodes. Various selection methods to visit and charge a group of sensor nodes in each round are proposed. The algorithms using the latter assumption persumes that a mobile charger has sufficient battery capacity to visit and charge all sensor nodes. In the forthcoming sections, these algorithms are discussed further.

### 2.4.2 Unlimited Battery Capacity

Xie et al. [62] define for the first time a renewable energy cycle, which corresponds to the energy level of a sensor node if it periodically changes over a range with a fixed minimum and maximum value, and it never falls below a threshold. The aim is to maximize the vacation time of the mobile charger while satisfying the renewable energy cycle of all sensor nodes. Mobile charger’s vacation time is the time when the current charging round finishes until the next charging round starts. The problem is formulated as a nonlinear optimization problem and the authors prove that the optimal solution is the sequence constructed using the shortest Hamiltonian path that
satisfies the renewable energy cycle. The proposed charging method in [62] is not scalable when the number of sensor node is increased. This motivates the authors to propose a multi-node charging solution [124], where a mobile charger charges multiple sensor nodes at each stop. The authors divide the network area into hexagonal cells where a mobile charger anchors at the cell center and charges all sensor nodes in its vicinity.

Zhao et al. [63] propose a Joint Mobile Energy Replenishment and Data Gathering (J-MERDG) algorithm to maximize network utility. J-MERDG finds a sequence of sensor nodes, called anchor points, to be charged to their maximum battery level by a mobile entity called SenCar. Each anchor point acts as a temporary sink. This means SenCar plays two roles: recharging and data collection. SenCars visiting tour length is limited by the required delivery time of collected data. Therefore, J-MERDG determines its tour by sorting sensor nodes in ascending order based on their residual energy and finding the maximum number of sensor nodes to be visited such that the required data delivery time bound is met. The major problem with J-MERDG is that it sorts sensor nodes according to their residual energy and does not consider the consumption rate of each node, which has a higher impact on the network lifetime.

2.4.3 Limited Battery Capacity

In practice, a mobile charger does not have unlimited battery capacity. To this end, Yang et al. [60] propose the Greedy-Plus (GP) charging algorithm, where the mobile charger has limited battery capacity. GP aims to minimize the difference between sensor nodes lifetime. It first sorts $K$ nodes in
ascending order; say \( \{n_1, n_2, n_3, \ldots, n_k\} \), where \( n_1 \) is the node with the least battery life. In the first step, GP considers the lifetime of node \( n_2 \) as the target lifetime for node \( n_1 \) and finds a charging tour to increase the lifetime of \( n_1 \) up to \( n_2 \). Then, if this target lifetime is achieved, the algorithm sets the target lifetime to \( n_3 \) for node \( n_1 \) and \( n_2 \) in the next step and so forth. In each step, GP checks \( J! \) permutation of a given path with \( J \) nodes \( 1 \leq J \leq K \). It then determines amongst \( K^2 K! \) tours the one that is able to meet the highest target lifetime. If a target lifetime is not achievable for a tour, GP resorts to binary search. A major limitation of GP is that it does not consider the shortest travel path. This implies that a mobile charger spends unnecessary amount of time travel between nodes. The other problem with GP is that it uses the binary search algorithm only when the target lifetime cannot be achieved. In scenarios where the target lifetime is achieved, GP does not use the remaining energy in a mobile chargers battery to increase lifetime further.

Li et al. [61] show that selecting a charging sequence is affected by the routing strategy. Bottleneck sensor nodes at the intersection of multiple data forwarding paths deplete their batteries sooner than other nodes even though they are recharged by the mobile charger. Hence, they propose a Joint Routing and Charging (J-RoC) scheme, where the data forwarding paths are changed periodically in each charging round based on the energy level of sensor nodes. The charging phase of J-RoC uses binary search to find the maximum achievable target lifetime with the constraint that the total energy used to recharge sensor nodes is within the mobile chargers battery capacity. The problem with J-RoC is that it does not consider the energy consumption
of the mobile charger due to travel.

Angelopoulos et al. [120] propose a local Adaptive Circular Traversal Strategy (ACTS) in order to reduce the communication overhead of a sensor node, and thereby, to increase its lifetime. A mobile charger selects its direction based on the information gathered during its visit to sensor nodes. Sensor nodes are placed in concentric circles around the sink. The mobile charger starts from the closest circle to the sink and moves towards the outer circle in order to charge sensor nodes in each circle. The amount of energy given to each sensor node is proportional to the mobile chargers residual energy.

Before charging a sensor node, the mobile charger records the current energy level of the sensor node. When charging all sensor nodes in one circle is completed, mobile charger determines its moving direction based on the energy level of the previously visited circles. If the energy level of latest circle is larger than the previous visited circle, the mobile charger changes its moving direction. Otherwise it continues in the same direction. Changing the direction means moving towards the outer circle if the move has been towards the inner circle and vice versa. The main problem with ACTS is that the amount of energy given to sensor nodes is independent of their energy consumption rate.

He et al. [121] propose an on-demand charging algorithm called Nearest-Job-Next with Preemption (NJNP), which is different from offline charging algorithms. That is, in offline charging models[62] [60] [61] [63], a set of nodes is selected for visit by a mobile charger in each charging round. In NJNP, a mobile charger selects the next sensor node after receiving requests from
sensor nodes. Each sensor node sends a charging request when its energy level falls below a threshold. Whenever the mobile charger receives a new charging request or completes charging a sensor node, among all requests, the mobile charger selects and charges the nearest sensor node that sends the query. The main problem with NJPN is that it does not consider energy consumption rate. Consequently, a request from a faraway sensor node with a high energy consumption rate may not get served and the node dies because of large number of received requests from low energy consumption rate nodes around the mobile charger.

Multiple mobile chargers are considered in [122], in which mobile chargers collaboratively charge each other. The objective is to maximize the charging time of sensor nodes and also to minimize the residual energy of mobile chargers upon returning to the sink position. The authors consider a chain topology and 100% energy transfer efficiency. In scenarios with two mobile chargers, a mobile charger first starts from the sink position and visits sensor nodes that are between the sink node and a point $P_1$. Here, point $P_1$ is at distance $L_1$ from the sink and is a RP for mobile chargers to meet each other. A second charger starts from the sink node and without visiting any sensor nodes goes directly to $P_1$. After being fully recharged by the first charger at $P_1$, the second charger visits sensor nodes that are at distance $L_2$ ($L_2 > L_1$) from the sink and goes back to $P_1$. Then the first mobile charger charges the second one so that both have sufficient energy to return to the sink position. The number of Rendezvous Points and their positions are determined by the charging algorithm and are dependent on the number of available mobile chargers. The main problem with this approach is its assumption of 100%
charging efficiency. In fact, as pointed out in [119], the maximum recorded wireless energy transfer efficiency is only 40% where the distance between two devices is less than two meters.

Chiu et al. [123] consider a scenario where sensor nodes are mobile and wireless charges are static. The authors consider a wireless patient health monitoring network that is used to regularly detect the blood pressure of patients with hypertension [125]. Patients carry sensor nodes that measure their blood pressure. The authors propose the Mobility-Aware Charging Deployment (MACD) algorithm to distribute the minimum number of static wireless chargers to fully charge sensor nodes. MACD converts a city map into a grid-based map, where each grid point is a possible location for a static charger. Each sensor node has a regular moving pattern that enables the MACD algorithm to calculate the number of sensor nodes seen at each candidate point. Using this information, MACD selects the minimum number of candidate points to place static chargers such that each sensor node is seen by at least one static charger.

2.5 Summary

This chapter has presented an in-depth survey and classification of coordination in WSANs, mobile sink and charger approaches. The coordination problem cuts across different layers of the protocol stack. Moreover, unlike WSNs, protocols must be developed according to a set of requirements that are distinct to WSANs.

Mobile sink approaches can be classified based on their applications: real-
time and delay-tolerant. In real-time applications, the main concern is to find the right feasible sites for a mobile sink and its sojourn time at each feasible site. In delay-tolerant applications, balancing between required packet delivery time and sensor node’s energy consumption is the main challenge.

Proposed methods that use wireless charging in WSNs can be grouped based on their battery capacity assumption as: unlimited and limited. In the unlimited battery capacity case, the main issue is to find the shortest visiting tour to charge all sensor nodes. In limited battery capacity case, the primary issue is to select a set of sensor nodes to visit and the amount of given energy to give them. Apart from that, it is unrealistic to assume that a mobile charger has unlimited battery capacity (see [62] [63]). GP [60], J-RoC [61], ACTS [120] and NJNP [121] are wireless charging algorithms that consider limited battery capacity. However, they have not considered the shortest path to visit sensor nodes. Intuitively, the energy saved from travel over the shortest path increases charging times.

In the next chapter, this thesis will outline an approach that considers the use of a mobile sink to collect data from a subset of nodes called rendezvous points. As a result, the approach reduces multi-hop communications, which in turn, mitigates the energy holes problem. After that, Chapter 4 explores the use of a mobile charger. This overcomes a fundamental limitation of WSNs/WSANs, the finite battery capacity of sensor nodes. Consequently, energy holes can be mitigated by recharging nodes with a high energy consumption rate.
Chapter 3

Weighted Rendezvous Planning

3.1 Introduction

This chapter addresses the delay aware energy efficient path problem. The main challenge is to identify the most suitable Rendezvous Points (RPs) for a mobile sink in order to minimize the energy consumed by sensor nodes during multi-hop communications whilst meeting a given packet delivery bound. It will be shown that the problem is NP-hard. A heuristic algorithm called Weighted Rendezvous Planning (WRP) is then proposed to determine the mobile sink’s trajectory and the set of RPs that optimize the energy consumption of sensor nodes. WRP is validated through extensive computer simulation. The results demonstrate that the trajectory and RPs generated by WRP enable a mobile sink to retrieve all data from sensor nodes within a given deadline whilst optimizing the energy consumed by sensor nodes.

3.2 Problem Formulation

One of the proposed approaches in the literature to mitigate energy holes in WSNs is to use a mobile sink [33] [50]. In this method, an autonomous mobile rover or sink travels within a network area and collects data directly from
each sensor node. Using this approach, sensor nodes no longer forward data to a base station. However, in a network with a large number of sensor nodes and strict delay tolerance such as like precision agriculture [20][5][21][22] (See Table 1.1), visiting all sensor nodes violates the required packet delivery time. To meet the delay bound constraint, RPs have been proposed by researchers in order to limit a mobile sink’s tour length [36] [34]. In this approach, a subset of sensor nodes are selected as RPs, and non-RP sensor nodes forward their data to the selected RPs. The mobile sink only visits RPs and collects buffered data from them. Selecting the most suitable RPs that minimizes energy consumption and meeting a given packet delivery bound is the main problem. In addition, selecting the set of RPs that result in uniform energy expenditure amongst sensor nodes in order to maximize network lifetime is a secondary problem. The next section formulates the problem succinctly.

3.2.1 Assumptions

The proposed solution makes the following assumptions:

1. The communication time between mobile sink and sensor nodes is negligible as compared to the travel time of mobile sink. Similarly, the delay due to multi-hop communications including transmission, propagation, and queuing delays are negligible with respect to the travel time of the mobile sink in a given data collection round.

2. Each RP node has sufficient storage to buffer all the collected data.

3. The stop time of mobile sink at each RP in order to collect the buffers data from RP is sufficient to drain all the stored data.
4. The mobile sink is aware of the location of each RP.

5. All sensor nodes are connected which means at least one data forwarding path exists between each two sensor nodes. Therefore there are no isolated sensor nodes.

6. Sensor nodes have a fixed data transmission range.

7. Each sensor node produces one data packet with the length of $b$ bits within the time interval of $D$ seconds.

3.2.2 Notation

A WSN can be modeled as a two-tuple graph $G(V, E)$, where $V$ is the set of homogeneous sensor nodes, and $E$ is the set of edges between the sensor nodes. An edge is a single hop data forwarding path between two sensor nodes which means both sensor nodes are in communication range of each other. If a sensor node $i$ sends $b$ bits to node $j$, its energy consumption is [66]:

$$E_{TX}(i, j) = b(\alpha_1 + \alpha_2 \times d_{i,j}^\gamma)$$  \hspace{1cm} (3.1)

where $d_{i,j}$ is the physical distance between sensor node $i$ and $j$ and $\alpha_1$ is the energy consumption index defining the power consumed by the transmission circuit to transmit one bit. $\alpha_2 d_{i,j}^\gamma$ is the energy consumption of the transmitter per bit and $\alpha_2$ is the energy consumption factor of the amplifier circuit. Here, $\gamma$ is the path loss exponent, which usually varies between 2 and 4, depending on the environment. Path loss is the power density reduction of
the electromagnetic wave when it propagates through the air. The power consumed by node $i$ to receive $b$ bits from node $j$ is,

$$E_{RX}(i, j) = b \times \beta$$ \hspace{1cm} (3.2)

where $\beta$ is a factor that represents the energy consumption per bit of the receiving circuit. As sensor nodes generate the same size data packets and have fixed data transmission range (assumption 6 and 7), therefore, $E_{TX}$ and $E_{RX}$ do not depend on the distance between sensor nodes and are identical for all sensor nodes.

The mobile sink node moves with a constant speed $v$. Hence, the maximum length of the travelled path $l$ is,

$$l_{max} = D \times v$$ \hspace{1cm} (3.3)

A mobile sink node starts its movement from a node $m_0 \in V$ and before time $D$, returns to its starting point. Each sensor node sends its generated data packets to the closest RP through multi-hop transmissions.

Let $H(i, M)$ be a function that returns the closest RP in terms of hop count to the sensor node $i$, where $M$ is the set of RPs. Specifically,

$$H(i, M) = \{ h_{i,m_j} | \forall m_k \in M, h_{i,m_j} \leq h_{i,m_k} \}$$ \hspace{1cm} (3.4)

where $h_{i,j}$ is the hop distance between node $i$ and $j$.

For each RP $m_i$, the proposed algorithm constructs a data forwarding tree $T_{m_i}$, comprising of the closest sensor nodes to that RP. The number of
data packets $NFD(i)$ that a sensor node $i$ forwards to the closest RP $m_i$ in each time interval $D$ is equal to its own generated data packet plus the number of its children in the data forwarding tree $T_{m_i}$. Specifically,

$$NFD(i) = C(T_{m_i}) + 1$$

where $C(T_{m_i})$ is a function that returns the number of descendants of sensor $i$ in $T_{m_i}$.

A summary of the notations used in this section is provided in Table 3.1.

3.2.3 Delay-aware Energy Efficient Travel Path (DEETP)

The objective is to find a tour $M = \{m_0, m_1, m_2, \ldots, m_n, m_0\}$ of length $l$, where $m_i \in V$, such that the length of tour $M(l)$ is not longer than $l_{\text{max}}$, and the network energy consumption for sending the collected data from sensor nodes to the tour $M$ is defined by $(E_{TX} + E_{RX}) \sum_{i \in V} H(i, M)$ is minimized within time interval $D$.

DEETP is NP-hard by a reduction from Travel Salesman Problem (TSP). The minimum energy consumption occurs when all sensor nodes are designated as an RP. This is because they do not incur any energy expenditure related to the forwarding of packets from other nodes. In this case, the goal is to determine whether there is a tour that is not longer than $l_{\text{max}}$. Henceforth, the next section proposes a novel heuristic method to approximate the optimal solution.
3. WEIGHTED RENDEZVOUS PLANNING

### Tab. 3.1: A summary of notations used in problem formulation.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G(V, E)$</td>
<td>Complete graph of a network</td>
</tr>
<tr>
<td>$V$</td>
<td>The set of sensor nodes</td>
</tr>
<tr>
<td>$E$</td>
<td>Set of edges corresponding to physical distance between sensor nodes</td>
</tr>
<tr>
<td>$n$</td>
<td>The number of sensor nodes in network, $n =</td>
</tr>
<tr>
<td>$H(i, M)$</td>
<td>Hop distance of node $i$ from the closest RP in $M$</td>
</tr>
<tr>
<td>$h_{i,j}$</td>
<td>Hop distance between node $i$ and $j$</td>
</tr>
<tr>
<td>$NFD(i)$</td>
<td>Number of data packets forwarded by node $i$</td>
</tr>
<tr>
<td>$v$</td>
<td>Speed of mobile sink</td>
</tr>
<tr>
<td>$l_{max}$</td>
<td>Maximum allowed tour length</td>
</tr>
<tr>
<td>$M$</td>
<td>Set of RPs</td>
</tr>
<tr>
<td>$m_i$</td>
<td>Sensor node $i$ which is in the tour</td>
</tr>
<tr>
<td>$E_{TX}$</td>
<td>Energy incurred when transmitting data</td>
</tr>
<tr>
<td>$E_{RX}$</td>
<td>Energy consumption for receiving data</td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td>Energy consumption factor per bit of transmitter circuit</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>Energy consumption factor per bit of amplifier circuit</td>
</tr>
<tr>
<td>$d_{i,j}$</td>
<td>Distance between node $i$ and $j$</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Path loss exponent</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Energy consumption factor per bit in receiver circuit</td>
</tr>
<tr>
<td>$D$</td>
<td>Maximum allowed packet delay</td>
</tr>
<tr>
<td>$T_{m_i}$</td>
<td>Data forwarding tree rooted at RP $m_i$</td>
</tr>
<tr>
<td>$C(i, T_{m_j})$</td>
<td>The number of children that node $i$ has in the data forwarding tree rooted at its corresponding RP $m_j$</td>
</tr>
<tr>
<td>$W_i$</td>
<td>Weight of sensor node $i$</td>
</tr>
</tbody>
</table>

### 3.3 Weighted Rendezvous Planning

Weighted Rendezvous Planning (WRP) algorithm preferentially designates sensor nodes with the highest weight as a RP. The weight of sensor node $i$ ($W_i$) is the product of the number of packets that it forwards $NFD(i)$ and
its hop distance from the closest RP on the tour $H(i, M)$:

$$W_i = NFD(i) \times H(i, M)$$  \hspace{1cm} (3.6)

Based on (3.6), a sensor node that is one hop away from a Rendezvous Point and has one buffered data packet, has the minimum weight. Accordingly, a sensor node that is far away from the selected RPs or has more than one packet in its buffer has a higher priority of being assigned as a Rendezvous Point (RP).

According to equations 3.1 and 3.2, the energy consumption is proportional to the hop count between source and destination nodes, and number of forwarded data packets. Hence, visiting the highest weighted node, will reduce the number of multi-hop transmissions, and thereby, minimizes the energy consumption. In addition, energy holes are more likely to occur in highly dense areas that have higher number of nodes. Hence, a mobile sink that preferentially visits highly dense areas will prevent the formation of energy holes in a WSN.

The operation of WRP is illustrated in algorithm 1. The input to the algorithm is $G(V, E)$, and the output produced by it is a set of RPs. Initially, WRP adds the fixed sink node as the first RP (line 6). Then in lines 9-16, it adds the highest weighted sensor node. In the next step, WRP calls TSP(), (see line 21), to calculate the cost of the tour. If the tour length is less than the required length $l_{max}$, the node selected in lines 9-16 remains as an RP (lines 22 to 31). Otherwise, the node is removed from the tour (lines 32 to 36). When a sensor node is assigned as a RP, the RPs that no
longer receives data packets from sensor nodes are removed from the tour (lines 27 to 31). This is because adding a sensor node to the tour may reduce the number of data packets directed to these RPs. Consequently, this step provides WRP with more opportunity to add other nodes into the tour. Note that the variable “removed” is used to guarantee that an RP will be deleted from the tour only once. If a removed RP is added to the tour for the second time, because its corresponding variable “removed” is true, it will not be removed from the tour again. By this way, all sensor nodes will be added to the tour when the required tour length for a mobile sink is bigger than the time to visit all sensor nodes.

The process through which WRP generates a travel tour for a mobile sink is illustrated in Figure 3.1. The maximum tour length is $l_{\text{max}} = 90m$. WRP starts by adding the sink node to the tour, $M = [Sink]$. Then a Shortest Path Tree (SPT) rooted at the sink node is constructed; (see Figure 3.1(a)). In the first iteration, WRP adds node 10 to the tour because it has the highest weight, yielding $M = [Sink, 10]$. As shown in Figure 3.1(b), the tour length of $M$ is smaller than the required tour length ($56 < 90$), implying that node 10 stays in the final tour (lines 22-31). In the second iteration, WRP recalculates the weight of sensor nodes because node 10 is now part of the tour. In this iteration, WRP selects node 6 as the next RP that has the highest weight. As shown in Figure 3.1(c), the tour length of $M = [Sink, 10, 6]$ is larger than the required tour length ($119 > 90$). Consequently, WRP removes node 6 from the tour $M = [Sink, 10]$ (lines 32-36).

In the third iteration, the weight of sensor nodes will not change because
Algorithm 1 Weighted Rendezvous Planning (WRP) algorithm.

1: Input $G(V, E)$, $l_{\text{max}}$
2: Output $M = (m_0, m_1, m_2, \ldots, m_n, m_0) = \text{NULL}$, where $m_i \in V \cup \text{Sink}$
3: Int $T_n = 0$, $W_{\text{max}} = 0$, $\text{flag} = 0$, $\text{RP} = -1$
4: Float $\text{cost} = 0$
5: Boolean $\text{mark}[n]$, $\text{removed}[n]$
6: $M = M \cup \text{Sink}$; $T_n +$
7: while $T_n \leq |V|$ do
8: $W_{\text{max}} = 0$; $\text{flag} = 0$
9: for $i \leftarrow 0$ To $|V|$ do
10: $\text{NFD}(i) = C(T_{m_i}) + 1$
11: end for
12: for $i \leftarrow 0$ To $|V|$ do
13: if not $\text{mark}(i)$ and $(\text{NFD}(i) \times H(i, M) > W_{\text{max}})$ then
14: $\text{RP} = i$; $W_{\text{max}} = \text{NFD}(i) \times H(i, M)$; $\text{flag} = 1$
15: end if
16: end for
17: if !$\text{flag}$ then
18: break;
19: end if
20: $\text{mark}($RP$) = \text{true}$; $M = M \cup \text{RP}$; $T_n +$
21: $\text{cost} = \text{TSP}(M)$
22: if $\text{cost} \leq l_{\text{max}}$ then
23: for $i \leftarrow 0$ To $|V|$ do
24: if $i \notin M$ then
25: $\text{mark}(i) = \text{false}$;
26: end if
27: if $C(T_{m_i}) == 0$ and $\text{mark}(i) == \text{true}$ and $\text{removed}[i] == \text{false}$ and $i \neq \text{RP}$ then
28: $\text{removed}[i] = \text{true}$; $\text{mark}(i) = \text{false}$; $M = M - i$; $T_n -$;
29: end if
30: end if
31: if $\text{cost} > l_{\text{max}}$ then
32: $M = M - \text{RP}$; $T_n -$;
33: $W_{\text{max}} = 0$; $\text{flag} = 0$
34: Go to line 12;
35: end if
36: end while
node 6 is not selected as an RP but it stays marked and will not be selected. WRP, however selects node 8 as it has the highest weight and is not marked (see Figure 3.1(d)). The TSP function returns 76 m for $M = [\text{Sink}, 10, 8]$ which is less than 90m. Therefore, node 8 is added to the tour. The process continues yielding a final tour of $M = [\text{Sink}, 8, 7, 10, 9]$ with a tour length of 81m, which is less than the required tour length (Figure 3.1(e)).

As shown in Figure 3.1, the final tour computed by WRP always includes sensor nodes as RPs that have more data packets to forward than other nodes. This ensures uniform energy consumption and mitigation of energy holes problem. This is the key advantage of WRP over Cluster Based (CB), Rendezvous Design for Variable Tracks (RD-VT) and Rendezvous Planning Utility-based Greedy (RP-UG). It will be shown in Section 3.4 that through
this property WRP can save up to 30% more energy than CB.

3.3.1 Analysis

One of the important properties of an algorithm is its time complexity. The time complexity for an algorithm is the execution time of that algorithm when called as a function. The time complexity is shown by $O$, which means the coefficients and lower order terms of input variables are excluded.

The time complexity of WRP is dependent on the number of times that WRP calls TSP solver to calculate a tour visiting all RPs. The worst case scenario is when all sensor nodes are marked but not selected as RP which means WRP will iterate for $|V|$ times to examine the possibility of adding nodes to a tour. After a node is selected as a RP, WRP again unmarks other sensor nodes and restarts the search process (lines 12 to 36). This implies that WRP uses the TSP solver for a maximum of $n^2$ times, where $n = |V|$. Hence, the time complexity of WRP is $O(n^2 \times O(TSP))$. Hence, using Christofides heuristic [126] TSP solver, which has a time complexity of $O(n^3)$, the resulting time complexity is $O(n^5)$. In the computer simulations conducted to verify WRP, the local search based heuristic TSP solver outlined in [127] is used.

WRP always finds a tour when there is at least one possible tour in the network as it checks the possibility of adding all sensor nodes to the tour. This is a significant property compared to CB and RD-VT because the latter two algorithms do not exhibit such characteristic. In CB, if the only possible tour consists of only the sink and a neighbor in the same cluster, CB will not be able to find this tour because two sensor nodes from the same cluster
cannot be in the final tour. As for RD-VT, it will not return any tour if the
distance of the first sensor node in depth first traversal of the Shortest Path
Tree (SPT) exceeds $l_{max}$.

Theorem 1, described below, indicates that when a mobile sink visiting
the most weighted nodes results in the least energy consumption as compared
to visiting any other nodes.

**Theorem 1.** Visiting sensor node $P$ with weight $w_p$ reduces energy consump-
tion more than visiting sensor node $Q$ with weight $w_q$, where $w_p > w_q$.

*Proof.* Recall that a sensor node $P$ forwards $NFD(P)$ data packets to its
closest RP. Hence, the energy consumption of sensor nodes on the path from
node $P$ to the closest RP is:

$$E_p = (E_{TX} + E_{RX}) \times (NFD(P) \times H(P, M))$$ (3.7)

where $E_{TX} + E_{RX}$ is the amount of consumed energy in each sensor node to
forward a data packet and $NFD(P) \times H(P, M)$ is the number of times that
the buffered data packets of sensor node $P$ will be forwarded until they reach
to the closest RP.

However, if sensor node $P$ becomes a RP, the energy consumption of the
network to deliver data packets of $P$ to the closest RP is zero. Similarly,
for sensor node $Q$ that forwards $NFD(Q)$ data packets to its closest RP, we
have,

$$E_q = (E_{TX} + E_{RX}) \times (NFD(Q) \times H(Q, M))$$ (3.8)

From Equ. 3.6, the weight of sensor node $P$ is $w_p = NFD(P) \times H(P, M)$
and the weight of sensor node $Q$ is $w_q = NFD(Q) \times H(Q, M)$. Since $w_p > w_q$, then according to (3.7) and (3.8), $E_p > E_q$ implying that selecting sensor node $P$ as an RP, which has a higher weight than $Q$, leads to less network energy consumption.

The next analysis shows the difference between WRP and the optimal solution. The following lemma is needed.

**Lemma 1.** Let $WRP_{op}$ be a version of WRP that uses an optimal TSP solver. If there is an optimal tour named $C$ with length $L_c \leq l_{\text{max}}$ comprising of all the sensor nodes as RP, then $WRP_{op}$ is guaranteed to identify tour $C$.

**Proof.** Assume there are $n$ sensor nodes in a WSN and tour $C = \{m_0, m_1, m_2, \ldots, m_n, m_0\}$, where $m_0$ is the sink node. Then,

$$L_c = \sum_{i=0}^{n-1} d_{m_i, m_{i+1}} + d_{m_n, m_0} \tag{3.9}$$

$WRP_{op}$, after picking the sink, will select node $m_{i=1}$ to include in the tour as it has the highest weight before running TSP(.) (see line 21 of Algorithm 1). The travel tour length will be less than $L_c$ as the tour connecting the set of nodes cannot be longer than the tour containing all the nodes by the triangle inequality. Hence, $WRP_{op}$ will add $m_i$ for $i = 2$ to $n$. $WRP_{op}$ then terminates when the number of RPs is equal to the number of sensor nodes $T_n = |V|$.

In Lemma-1, the requirement of an optimal TSP solver can be relaxed if we assume that $\sum_{i=0}^{E} \epsilon_i \leq l_{\text{max}}$ which implies that the sum of all distances between all sensor nodes is less than the required tour length.
It can be intuitively argued that the maximum difference in energy consumption occurs when the final tour returned by $WRP_{op}$ is not composed of any sensor nodes while the optimal tour visits all sensor nodes. However, as per Lemma-1, this argument does not hold.

**Theorem 2.** Assume a sensor node $P$ that has the longest hop distance from the sink and the average hop distance between sensor nodes and sink is $k$, then the maximum difference between the network energy consumption of $WRP_{op}$ and the optimal is within $\frac{2\times K \times (|V| - 1) + 1}{|V| + 2}$.

**Proof.** The network energy consumption when the mobile sink visits just sensor node $P$ is,

$$E_{Network(p)} = (E_{TX} + E_{TX} \times ((|V| - 1) \times k)) + (E_{RX} \times ((|V| - 1) \times k)) \quad (3.10)$$

On the other hand, the minimum amount of energy consumed by visiting all sensor nodes except node $P$ is

$$E_{Network(|V| - 1)} = E_{TX} \times (|V| + 1) + E_{RX} \quad (3.11)$$

This means sensor node $P$ has to send all its data packets to the closest RP while other sensor nodes send their data packets directly to the mobile sink. From Equ. 3.10 and 3.11, the ratio of energy consumption in WRP in comparison to the optimal model is

$$Ratio = \frac{E_{TX} \times (1 + (|V| - 1) \times k) + E_{RX} \times ((|V| - 1) \times k)}{E_{TX} \times (|V| + 1) + E_{RX}} \quad (3.12)$$
If we consider $E_{TX} \approx E_{RX}$, Equ. 3.12 is equal to

$$Ratio \approx \frac{2 \times K \times (|V| - 1) + 1}{|V| + 2}$$  \hspace{1cm} (3.13)

### 3.4 Validation

WRP is compared against three existing methods with the same objective as the proposed method, including CB, RD-VT and RP-UG, using a custom simulator written in C++\footnote{The developed simulator is available upon request.}. In the validation, a connected WSN is considered in which nodes are placed uniformly over a sensor field of size 200 x 200 m$^2$. It is important to note that interconnecting disconnected sensor nodes using a mobile node is a well known and a separate problem. Although WRP can be also applied to interconnect disconnected nodes if the required delivery time for data packets is greater than the shortest travel tour to visit all sensor nodes. The energy holes are more likely to form when nodes are distributed uniformly [128]. Experimental results in [129] demonstrate that if sensor nodes are distributed uniformly, up to 90% of residual energy is unused when the first sensor node runs out of energy. Similar to RD-VT, CB and RP-UG, uniform distribution is deployed.

Based on [105], the radio parameters are set as per the CC1000 radio, which is used by Mica2 Motes [130] or TelosB [131] nodes. Each sensor node generates one data packet every $T$ Sec, which is then forwarded to an RP via a shortest path tree. Nodes are aware of the mobile sink’s movement and
3. WEIGHTED RENDEZVOUS PLANNING

hence, its arrival time. In each experiment, the network energy consumption is recorded every $T$ Sec. It is assumed that there are 200 sensor nodes in WSN, adequate for most applications (see Table 1.1).

In order to measure network lifetime, all sensor nodes have a fully charged battery with 100 j of energy. Other parameters of the network are summarized in Table 3.2. The velocity of the mobile sink is set to 1 m/s. Moreover, the mobile sink visits each RP. Given a transmission range of 20 m, which is feasible for Mica2 [130] or TelosB [131] nodes, the mobile sink will be in a sensor node’s transmission range for 20 seconds. Assuming a data transmission rate of 40 Kbps, each sensor node will be able to send 3,413 data packets of length 30 bytes to the mobile sink in 20 seconds. This means that the mobile sink has sufficient time to drain the buffer of all sensor nodes even when there are 200 sensor nodes. To reduce the run time of RP-UG, $L_0$ is set to 20 m, which corresponds to the transmission range of sensor nodes.

The imbalance in energy consumption is measured by estimating the standard deviation (SD) of sensor nodes energy consumption rates. A large standard deviation indicates that some parts of WSN is likely to exhaust its energy faster than other parts. The metric SD is calculated as follows,

$$SD = \sqrt{\frac{\sum_{i \in V} (EN[i] - \mu)^2}{|V|}}$$

where $EN[i]$ is the energy consumption of node $i$, $V$ is the set of sensor nodes and $\mu$ is the average energy consumption of sensor nodes.

Two scenarios are considered in the validation involving Shortest Path Tree (SPT) and Steiner Minimum Tree (SMT) for the RD-VT model, RP-UG
3. WEIGHTED RENDEZVOUS PLANNING

*Tab. 3.2: Simulation parameters.*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum allowed packet delay (D)</td>
<td>100 to 300 seconds</td>
</tr>
<tr>
<td>Number of sensor nodes (n)</td>
<td>7 to 200</td>
</tr>
<tr>
<td>Mobile sink speed (v)</td>
<td>1 m/s</td>
</tr>
<tr>
<td>Sensor nodes transmission range</td>
<td>20 m</td>
</tr>
<tr>
<td>Packet length (b)</td>
<td>30 bytes</td>
</tr>
<tr>
<td>Consumed energy in transmitter circuit</td>
<td>42mW</td>
</tr>
<tr>
<td>Consumed energy at the receiver circuit</td>
<td>29mW</td>
</tr>
<tr>
<td>Sensor node’s battery</td>
<td>100 J</td>
</tr>
</tbody>
</table>

and WRP. SPT is rooted at base station and constructed over the network graph \(G\) to find the shortest hop path from each sensor node to the base station. SMT is rooted at base station and constructed over the network graph \(G\) to find a hop path from each sensor node to the base station such that the sum of all hop paths is minimized. In the SMT, extra intermediate nodes may be added to the graph in order to reduce the length of the hop paths. The new added nodes in SMT known as Steiner points. In WRP, Steiner points are treated as real nodes. This means Steiner points have a weight and are not replaced with the real sensor nodes in the final tour.

Two sets of experiments are carried out. Initially, the number of nodes is limited to 20 and WRP is compared against optimal WRP which yields the optimal tour with 2.5 minutes as the required tour length. In the second experiment, the number of nodes is increased to 200 and WRP is compared against RD-VT, RP-UG and CB with tour length of 5 minutes. In all experiments, each node has a unique ID number and the node with the highest ID is designated as the sink node. The results are an average of ten simulation runs over different topologies.
3. WEIGHTED RENDEZVOUS PLANNING

3.4.1 Performance under SPT

Figure 3.2 shows the energy consumption of sensor nodes for WRP versus brute force. The brute force is the optimal version of WRP which goes throughout all the possible tours for mobile sink and finds a tour with the maximum weight and number of nodes. The optimal WRP uses the brute force TSP solver for calculating the tour length. Both algorithms yield higher energy consumption when the number of sensor nodes increases as the length of the path for forwarding from sensor nodes to RPs increases. The energy consumption of WRP is very close to the brute force approach. More specifically, brute force outperforms WRP only by 5%. The standard deviations of energy consumption for different number of sensor nodes in brute force and WRP-SPT are illustrated in Figure 3.3. A small standard deviation implies uniform energy consumption and longer network lifetime. The performance of WRP is just 16% less than the optimal or brute force approach. This is because, the sensor nodes of WRP that forward more data packets and cause more multi-hop transmissions than other sensor nodes are likely to be designated as an RP.

Figure 3.4 shows the energy consumption for WRP, CB, RD-VT and RP-UG with a large number of sensor nodes. RD-VT leads to the highest energy consumption because of its pre-order traversal of SPT and long data forwarding paths from sensor nodes to the RPs. WRP recorded 47% reduction in energy consumption as compared to RD-VT. CB has better performance than RD-VT because in its finalization process, if the required delivery time is not violated, it replaces the selected RP in each cluster with a node closer
Fig. 3.2: Network energy consumption between brute force and WRP.

to the cluster head in order to reduce the number of multi-hop transmissions. CB’s performance is 28% better than RD-VT in terms of energy consumption. Recall that in Section 2.3.3 CB does not consider node density or hop counts when selecting RPs. As a result, WRP achieves a 10% reduction in energy consumption as compared to CB. RP-UG and WRP have nearly the same performance with only 6% reduction of energy consumption in compared to RP-UG.

The ability of the four algorithms to uniformly distribute energy consumption is shown in Figures 3.5 and 3.6. WRP distributes energy more uniformly than the other approaches, specifically, 12% more than RP-UG,
28% more than CB, and 53% better than RD-VT. Similarly, RP-UG does not aim to balance the energy consumption rate of sensor nodes. RP-UG adds sensor nodes that are close to the sink as RPs, which may not necessarily have the highest energy consumption rate. Moreover, even though RP-UG considers nodes that are on many routing paths, WRP preferentially selects nodes with high energy consumption rate and hop count from the sink. As illustrated in Figure 2.6, the random cluster head selection process of CB causes non-uniform energy consumption. Moreover, two sensor nodes from the same cluster cannot be in the final tour. Hence, when there are a large number of sensor nodes, energy holes are likely to occur around the
RP for a given cluster. In contrast, WRP avoids this scenario by having the mobile sink visiting highly populated sections of a WSN, thus reducing the number of multi-hop transmissions. In RD-VT, long data forwarding paths from sensor nodes to RPs results in non-uniform energy consumption and 25% reduction in network lifetime as compared to CB.

The computer simulation applies the algorithms to a network of 110 sensor nodes with data packets having delivery time ranging from 100 to 300 seconds. Figures 3.7 and 3.8 illustrate the network energy consumption and network lifetime for WRP, CB, RD-VT and RP-UG. Consistent with the result shown in Figure 3.4, WRP yields the best performance amongst all the
algorithms. The energy consumption for RD-VT is reduced by 21% when the required packet delivery time is changed from 100 to 300 seconds while WRP, RP-UG and CB experience 41%, 33% and 37% reduction in their energy consumption, respectively. WRP exhibits a superior performance compared to other algorithms even with small packet delivery times. This is due to the ability of WRP to add the node with the highest weight first.

Finally, the execution time for each algorithm is shown in Figure 3.9. For RP-UG, $L_0$ is set to 20$m$; i.e., the transmission range of sensor nodes. This is the maximum possible value for $L_0$ because otherwise edges bigger than $L_0$ are split into edges with length $L_0$. Virtual nodes are then added as
3. WEIGHTED RENDEZVOUS PLANNING

necessary to connect these new edges. Consequently, this process, depending on the value of $L_0$, increases run time significantly. Even with $L_0$ set to 20 m, the running time of RP-UG is six times larger than WRP, 36 times more than CB and 72 times longer than RD-VT. This is because, in each iteration, RP-UG calculates the utility of each sensor node by calling a TSP solver. RD-VT has the lowest running time because it only calls the TSP solver once in each iteration.

3.4.2 Performance under SMT

Let’s consider application of SMT in RP-UG, RD-VT and WRP. This tree is constructed using the function proposed in [132]. Specifically, the Steiner
tree function uses the principal of an equilateral triangle, a circle, and a line to construct a Steiner point for a set containing three points on the Minimum Spanning Tree (MST). When an SMT is formed, there are two types of nodes. The first type corresponds to real sensor nodes, and other type corresponds to the new added nodes which are simply physical positions with no sensor nodes, so called Steiner points or virtual nodes.

For each Steiner point in WRP, a neighbor set is assigned after the formation of SMT. In RP-UG, RD-VT and WRP, Steiner points that are not in the final tour are deleted from the SMT. On the other hand, Steiner points

Fig. 3.7: Network energy consumption for WRP, CB, RD-VT and RP-UG under different required delivery times for data packets.
that are part of a tour are called virtual RPs and are handled in the following manner. In RD-VT and RP-UG, virtual RPs are replaced with the closest physical sensor nodes. In WRP, when a sensor node notices that the next hop destination for its data packets is a virtual RP, it stores its data until the mobile sink arrives at the virtual RP’s position. Upon arrival, the sensor node forwards its data to the mobile sink.

Figure 3.10 shows a comparison between WRP, RP-UG, RD-VT and CB when SMT is used for data forwarding purpose. The results show that SMT has a better performance than SPT as a Steiner point in the shortest interconnection points between neighboring nodes. As a result, the total length
of SMT in comparison to SPT is shorter. This causes RD-VT-SMT and RP-UG-SMT to visit more RPs, and causes nodes to have 29% and 8% less energy consumption than RD-VT-SPT and RP-UG-SPT, respectively. Moreover, RD-VT-SMT outperforms CB. Because of the shortest interconnection points between neighboring nodes in SMT, WRP-SMT conserves 13% more energy than WRP-SPT and 22% more than CB. WRP also has 24% better performance in reducing network energy consumption than RD-VT, and 14% better than RP-UG when SMT is deployed. This is because WRP does not replace virtual RPs with the closest physical sensor nodes. Instead, as mentioned before, a mobile sink visits virtual RP positions and collects data from nearby sensor nodes.

Fig. 3.9: Simulation time for WRP, CB, RD-VT and RP-UG.
Figures 3.11 and 3.12 illustrate the difference between the energy consumption of sensor nodes and the network lifetime for SMT experiments. In particular, the recorded standard deviation of sensor node’s energy consumption rates is less than when SPT is deployed. This is due to the presence of Steiner points in the final tour that do not receive data from other sensor nodes. Instead, these points are visited by the mobile sink. In comparison, when using SPT, RPs have a higher energy consumption as compared to other sensor nodes. However, for the SMT case, when virtual RPs are in the final tour, there are fewer RPs. Hence, the number of actual nodes that act as RPs is reduced and the energy consumption is decreased accordingly.
Moreover, the energy consumption in the sensor nodes is distributed uniformly. For these reasons, the standard deviation of sensor node’s energy consumption rates for WRP-SMT is 16% less than WRP-SPT and 39% less than CB, as illustrated in Figure 3.11. The standard deviation of sensor node’s energy consumption rates for RD-VT-SMT is 28% less than RD-VT-SPT, and for RP-UG-SMT it is 5% less than RP-UG-SPT. This is because visiting more RPs leads to shorter data forwarding paths and thereby, better network lifetime. The difference between the standard deviations of sensor node’s energy consumption rates for WRP-SMT and RD-VT-SMT is 44%, and between WRP-SMT and RP-UG-SMT, 22% less than the results recorded in SPT experiments. This confirms that the better performance produced by WRP in the SMT scenario is due to the use of virtual RPs.

3.5 Conclusion

The main focus of this chapter was on the application of a mobile sink to address the energy holes problem. The solution, called WRP, controls the movement of a mobile sink by selecting the set of RPs such that the energy expenditure of sensor nodes is minimized and uniform. Moreover, WRP considers the required data packet delivery time. The validation of the algorithm carried out via computer simulation, indicates that WRP-SMT reduces the energy consumption of the tested WSNs by 22% as compared to CB. In terms of the difference between sensor nodes energy consumption, the results show that WRP uniformly distributes energy consumption 39% and 44% better than CB and RD-VT respectively.
A key assumption made in this chapter is that sensor nodes have finite battery and consequently the corresponding WSN has finite lifetime. As will be shown in Chapter 4, WSNs can potentially have perpetual lifetime if a mobile sink is equipped with a wireless charger that supplies sensor nodes with energy periodically.
Fig. 3.12: Network lifetime for WRP, CB, RD-VT and RP-UG in SMT scenarios.
Chapter 4

Binary Search Wireless Charging

4.1 Introduction

This chapter focuses on wireless recharging in WSNs. In particular, it investigates the use of an autonomous mobile rover/robot that recharges the batteries of sensor nodes. The main problems addressed include the order in which sensor nodes are visited, the recharging time and the distance travelled by the mobile rover. The wireless charging problem is formulated as an Integer Linear Program (ILP) with the objective of maximizing network lifetime and demonstrating that it is equivalent to the well-known NP-hard Capacitated Vehicle Routing Problem (CVRP). A heuristic method called Binary Search Wireless Charging (BSWC) is then proposed in which the rover preferentially visits sensor nodes with the shortest lifetime. The validity of the algorithm is proved mathematically and sufficient conditions ensuring infinite network lifetime are derived, with the assumption that there is no other failures. BSWC is validated via computer simulation and the results demonstrate that BSWC increases network lifetime significantly more than current wireless charging algorithms.
4.2 Problem Formulation

Similar to Chapter 3, a WSN is modeled as a two tuple $G(V, E)$, where $V$ is the set of homogeneous sensor nodes and $|V| = N$ where $N$ is the number of sensor nodes. Variable $E$ is the set of edges between nodes in $V$. The maximum battery capacity of sensor nodes is denoted as $B_s$ and that of the mobile charger as $B_m$. Each sensor node $n_i$ has an energy consumption rate of $r_i$ that is attributed to sending and receiving data packets. It is assumed that the energy consumption due to sensing tasks is negligible when compared with the energy consumed for sending and receiving data packets.

A mobile charger consumes $e_{tx}$ Joules per second ($J/s$) to charge a sensor node and consumes $e_{mv}$ $J/s$ when moving at a constant speed of $v$. As reported in [119], a 40% charging efficiency between a wireless charger and receiver node located at 2 m away is possible. Therefore, the amount of energy that is captured by a sensor node is $e_{rx}$ $J/s$ and is calculated based on Equ. 4.1, where $\eta$, e.g., 40%, is the charging efficiency,

$$e_{rx} = \eta \times e_{tx}$$

(4.1)

In a charging round $p$, the mobile charger starts travelling from the base station or depot. After visiting and charging sensor nodes, it returns to the depot to recharge its battery. In other words, in round $p$, the mobile charger visits a sequence of nodes $S_p = \{n_0, n_1, n_2, \ldots, n_N, n_0\}$, where $n_0$ is the base station and $n_i \in V$ are sensor nodes on the tour. In round $p$, a sensor node
i is charged for $t_{i,p}$ seconds and the total length of charging round $p$ is $T_p$,

$$T_p = \sum_{i=1}^{N} t_{n_{i,p}} + \frac{L_p}{v} + t_c$$

(4.2)

where $L_p$ is the tour length of the charging round $p$ and $t_c$ is the recharging time of the mobile charger’s battery when it is at the depot. In other words, $t_c$ is the vacation time for the mobile charger to recharge its battery for the next charging round. In this study, $t_c$ is assumed to be constant.

The residual energy of node $i$ at the start of a charging round $p$ is denoted as $e_{i,p-1}$; that is, the remaining energy from the previous round. This information can be obtained easily via multi-hop communications. At the end of round $p$, the residual energy is,

$$e_{i,p} = (e_{i,p-1} + t_{i,p} \times e_{rx}) - (T_p \times r_i)$$

(4.3)

where $t_{i,p} \times e_{rx}$ is the amount of energy captured by sensor node $n_i$ and $T_p \times r_i$ is the amount of energy expended waiting for the mobile charger to arrive plus the time for it to travel back to the depot.

The lifetime of sensor node $n_i$ at the start of a charging round $p$ as $z_{i,p-1}$ at the end of round $p$, the lifetime is,

$$z_{i,p} = \frac{e_{i,p}}{r_i} = z_{i,p-1} + \frac{t_{i,p} \times e_{rx}}{r_i} - T_p$$

(4.4)

Given $z_{i,p-1}$, the network lifetime is defined as follows. Consider a sorted sequence of sensor nodes before the charging round $p$ starts as $C_p = \{n_1, n_2, n_3, \ldots, n_N\}$, where $z_{i,p-1} \leq z_{i+1,p-1}$. Then the network lifetime is $Z = z_{1,p-1}$ as node $n_1$.
has the lowest lifetime among other sensor nodes and will die first. If all
sensor node’s battery is charged in $C_p$ up to $B_s$, then the sensor node with
the maximum energy consumption rate $r_{max}$ will die first. Thus the aim
is to increase the lifetime of the sensor node with $r_{max}$, denoted as $Z_{max}$.
Consequently, the maximum achievable network lifetime for WSN $G(V,E)$
in charging round $p$ is $Z_{max}$. To increase the lifetime of sensor node $n_j$ up to
$Z_{max}$ in charging round $p$, we need to charge node $n_j$ for $t_{j,p}$ time. This is
calculated as follows,

$$z_{j,p} = Z_{max}$$ \hspace{1cm} (4.5)

$$\frac{e_{j,p-1}}{r_j} + \frac{t_{j,p} \times e_{rx}}{r_j} - T_p = \frac{B_s}{r_{max}} - T_p$$ \hspace{1cm} (4.6)

$$t_{j,p} = \frac{(B_s \times \frac{r_j}{r_{max}}) - e_{j,p-1}}{e_{rx}}$$ \hspace{1cm} (4.7)

The charging problem in each round $p$ can be modeled as an ILP, where the
objective is to find the shortest travel tour for the mobile charger in charging
round $p$, $S_p = \{n_0, n_1, n_2, ..., n_N, n_0\}$, that allows all sensor nodes to be charged
up to $Z_{max}$. Formally the problem is (see Table 4.1 for a summary of each
notation),

$$\text{Minimize} \sum_{k=0}^{N} \sum_{j=0}^{N} c_{k,j}x_{k,j}$$ \hspace{1cm} (4.8)

$$e_{tx} \sum_{i=1}^{N} + e_{mv} \sum_{k=0}^{N} \sum_{j=0}^{N} c_{k,j}x_{k,j} \leq B_m$$ \hspace{1cm} (4.9)

$$t_{i,p} = \frac{(B_s \times \frac{r_i}{r_{max}}) - e_{i,p-1}}{e_{rx}} \quad \text{for} \quad i = 1, ..., N$$ \hspace{1cm} (4.10)

$$\sum_{j=0}^{N} x_{j,i} = \sum_{k=0}^{N} x_{i,k} \quad \text{for} \quad i = 1, ..., N$$ \hspace{1cm} (4.11)
4. BINARY SEARCH WIRELESS CHARGING

\[ u_0 = 1 \quad 2 \leq u_i \leq n \quad \text{for } i = 1, ..., N \tag{4.12} \]

\[ u_i - u_j + 1 \leq n(1 - x_{i,j}) \quad \forall i \neq 0, \forall j \neq 0 \tag{4.13} \]

The objective function, as defined by Equ. 4.8, minimizes the travel distance, and thereby, ensuring as much of the mobile charger’s remaining energy can be used for charging. The equation \( \sum_{k=0}^{N} \sum_{j=0}^{N} c_{k,j} x_{k,j} \) is the “cost” of the charging tour length where \( c_{k,j} \) is the physical distance between node \( n_k, n_j \) and \( x_{k,j} \) is a binary variable that is equal to one if node \( n_k \) and \( n_j \) are charged in tour \( p \), otherwise it is zero. Constraint 4.9 ensures that the amount of energy spent by the mobile charger for charging and travelling is bounded by \( B_m \). Constraint 4.10 ensures that a sensor node’s lifetime is increased to \( Z_{\text{max}} \). Constraint 4.11 guarantees that each sensor node is visited once by mobile charger. Finally constraints 4.12, 4.13 are the Miller-Tucker-Zemlin (MTZ) subtours elimination constraints that prevent the occurrence of disjoint loops or subtours.

Given an instance of the ILP formulation, consider the WSNs with three sensor nodes depicted in Figure 1.2, where the distance between nodes are shown by solid lines. Node-0 is the sink, and for each sensor node, the residual energy and energy consumption rate is denoted as \( (e_{1,p-1}, r_1) \); e.g., for sensor node 1, is \( (3J, 0.05mJ/s) \). The sensor node’s battery capacity is \( B_s = 10J \). The mobile charger’s battery capacity is \( B_m = 50J \) and its energy consumption rate due to travel is \( e_{mv} = 2J/m \) and transmitted energy for charging is \( e_{tx} = 1J/s \). Node-3 has the maximum energy consumption rate \( r_{\text{max}} = 0.1mJ/s \), and the maximum achievable network lifetime is \( Z_{\text{max}} = 27.7h \).
Tab. 4.1: A summary of notation.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G(V,E)$</td>
<td>Complete graph of a network</td>
</tr>
<tr>
<td>$V$</td>
<td>The set of sensor nodes</td>
</tr>
<tr>
<td>$E$</td>
<td>Set of actual distance between sensor nodes</td>
</tr>
<tr>
<td>$N$</td>
<td>The number of sensor nodes in network; $N =</td>
</tr>
<tr>
<td>$B_s$</td>
<td>Battery capacity of sensor node</td>
</tr>
<tr>
<td>$B_m$</td>
<td>Battery capacity of the mobile charger</td>
</tr>
<tr>
<td>$r_i$</td>
<td>Energy consumption rate of sensor node $i$</td>
</tr>
<tr>
<td>$v$</td>
<td>Speed of mobile charger</td>
</tr>
<tr>
<td>$e_{tx}$</td>
<td>Transmitted energy by mobile charger $J/s$</td>
</tr>
<tr>
<td>$e_{rx}$</td>
<td>Received energy by sensor node $J/s$</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Charging efficiency</td>
</tr>
<tr>
<td>$e_{mv}$</td>
<td>Consumed energy by mobile charger for moving</td>
</tr>
<tr>
<td>$t_{i,p}$</td>
<td>Assigned charging time to node $i$ in charging round $p$</td>
</tr>
<tr>
<td>$T_p$</td>
<td>Time length of charging round $p$</td>
</tr>
<tr>
<td>$L_p$</td>
<td>The length of charging round $p$</td>
</tr>
<tr>
<td>$t_c$</td>
<td>Fixed mobile charger’s battery recharging time</td>
</tr>
<tr>
<td>$e_{i,p}$</td>
<td>Residual energy of sensor node $i$ at the end of charging round $p$</td>
</tr>
<tr>
<td>$z_{i,p}$</td>
<td>Lifetime of sensor node $i$ at the end of charging round $p$</td>
</tr>
<tr>
<td>$Z$</td>
<td>Network lifetime</td>
</tr>
<tr>
<td>$Z_{max}$</td>
<td>The maximum achievable network lifetime</td>
</tr>
<tr>
<td>$r_{max}$</td>
<td>Maximum sensor node’s energy consumption rate in the network</td>
</tr>
<tr>
<td>$x_{i,j}$</td>
<td>Binary variable that is one if node $i$ and $j$ are charged in current charging round</td>
</tr>
<tr>
<td>$c_{i,j}$</td>
<td>The physical distance between node $i$ and $j$</td>
</tr>
</tbody>
</table>

The shortest charging tour for the mobile charger starts from the sink position and visits all sensor nodes; i.e., $S_p = \{n_0, n_1, n_2, n_3, n_0\}$. Hence,

$$x_{0,1} = x_{1,2} = x_{2,3} = x_{3,0} = 1$$  \hspace{0.5cm} (4.14)
As shown in Figure 1.2, the total travel length for charging sequence $S_p$ is 13 m. Based on Equ. 4.14, constraints 4.11, 4.12 and 4.13 of the ILP are satisfied. The charging time of each sensor node to charge up to $Z_{\text{max}}$ based on Equ. 4.10 where $\eta$ is 40% is,

$$t_{1,p} = 5 \quad t_{2,p} = 5 \quad t_{3,p} = 10$$  \hspace{1cm} (4.16)$$

The total charging time is 20s. Hence the mobile charger spends $20 + 13 \times 2 = 46 J$ to visit and charge sensor nodes in $S_p$, which is less than $B_m$, and satisfies constraint 4.9. As illustrated in Figure 1.2, there is no other possible tour for the mobile charger that can visit and charge sensor nodes up to $Z_{\text{max}}$. Therefore, finding the shortest travel tour and $Z_{\text{max}}$ is thus the optimal target lifetime for WCP.

The key decision variables of the ILP are the charging time $t_{i,p}$ for each sensor node such that its lifetime can be increased to $Z_{\text{max}}$. This problem is similar to the well-known Capacitated Vehicle Routing Problem (CVRP)
where the charging time of each sensor node can be considered as a customer’s demand and the mobile charger’s remaining energy $B_m$ is the vehicle’s capacity. The customer’s demand must be less or equal to the vehicle’s capacity and the goal is to find the tour with minimum length through which all customers are visited and their goods are delivered - an NP-hard problem [133]. Hence, for this problem, a novel heuristic, called BSWC, is proposed to approximate the optimal solution.

### 4.3 Heuristic - BSWC

The proposed heuristic method is called Binary Search Wireless Charging (BSWC). A mobile wireless charger starts its tour from the base station position and preferentially visits sensor nodes with the shortest lifetime. Before the charging round $p$ starts, BSWC determines a sorted set of sensor nodes: $C_p = \{n_1, n_2, n_3, \ldots, n_N\}$, where $z_{i,p-1} \leq z_{i+1,p-1}$. From this set, BSWC will charge the first $K$ nodes. Specifically, BSWC finds the shortest travel path between these $K$ nodes and charges them to $z_{K,p-1} \leq Z_{\text{max}}$.

The algorithm starts with $K < N$ nodes as it is impractical to visit and charge all $N$ sensor nodes up to $Z_{\text{max}}$ in large WSNs. Given $K$ sensor nodes, BSWC will gradually decrease $K$ to find the best number of sensor nodes that the mobile charger can practically visit and charge to $z_{K+1,p-1}$. Specifically, if the required energy to visit and charge $K$ sensor nodes up to $z_{K+1,p-1}$ is more than $B_m$, BSWC reduces $K$ to $K - 1$ and considers a target lifetime of $z_{K-1,p-1}$ for the first $K - 1$ sensor nodes of $C_p$. This procedure continues until BSWC finds a set of $K$ sensor nodes that it can visit and increase their
4. BINARY SEARCH WIRELESS CHARGING

lifetimes to \( z_{K+1,p-1} \); i.e., the network lifetime becomes \( z_{K+1,p-1} \).

After finding a feasible set of \( K \) sensor nodes to visit and charge up to \( z_{K+1,p-1} \), the mobile charger may be left with energy. BSWC distributes this energy using binary search algorithm. The upper bound and lower bound of the binary search algorithm are the target lifetime where all \( K \) sensor nodes are charged up to \( B_s \) as an upper bound, and \( z_{K+1,p-1} \) as the lower bound. The output of the binary search algorithm is \( z_{op,K} \) where \( z_{K+1,p-1} \leq z_{op,K} \).

Algorithm 2 outlines the pseudo code for BSWC. The inputs are \( G(V,E) \), \( K \) and the set of sensor node’s lifetime \( C_p \). The outputs include (i) \( M \), the set of sensor nodes to be visited, and (ii) \( A \), a set that records the charging times of sensor nodes in \( M \). To find the charging tour \( p \), BSWC first starts with \( K \) sensor nodes and if \( K = N \) then the target lifetime \( z_{op,K} \) is set to \( Z_{max} \) otherwise \( z_{op,K} \) is set to \( z_{K+1,p-1} \) - (lines 6-15). The first \( K \) sensor nodes of \( C_p \) are added to \( M \) to be charged - (lines 16-18). BSWC uses the \( TSP(M) \) function to find the shortest path that visits \( |M| \) nodes, and calculates the mobile charger’s energy consumption to travel through that path - (line 19).

The variable \( b_m \) denotes the leftover energy of the mobile charger’s battery after taking into account the energy spent on travel - (line 19). If \( b_m \) is less than or equal to zero - (line 20), it means that the mobile charger does not have sufficient energy to charge sensor nodes. This causes BSWC to reduce the number of nodes to be visited - (lines 53 to 55), and goes to line 4. On the other hand, if \( b_m \) is greater than zero but it is not sufficient to charge sensor nodes up to \( z_{K+1,p-1} \) - (line 33), BSWC reduces the number of nodes to be visited - (lines 53 to 55), and goes to line 4.

When \( b_m \) is greater than zero, BSWC determines the upper and lower
bounds for the binary search algorithm in line 21. The lower bound low is 
$z_{K+1,p-1}$ and the upper-bound up is $z_{N+1}$, which is the maximum achievable 
lifetime for a sensor node with the lowest energy consumption rate. If the 
target lifetime $z_{N+1}$ can be achieved, it means all sensor nodes in $M$ are 
charged up to $B_s$. In each iteration, the binary search algorithm outputs a 
new $z_{op,K}$ value and calculates the charging time for each sensor node in $M$ 
to increase their lifetime up to $z_{op,K}$ - (lines 24 to 31). If the required energy 
to charge $|M|$ sensor nodes up to new $z_{op,K}$ is greater than $b_m$ - (line 32), then 
binary search reduces the upper-bound by setting $up = z_{op,K}$ and finds a new 
$z_{op,K}$ between low and up - (line 34). Otherwise, if $b_m$ is sufficient to charge 
sensor nodes in $M$ up to $z_{op,K}$, the binary search increases the lower-bound 
to $low = z_{op,K}$ and finds a new $z_{op,K}$ between low and up - (lines 38 to 48). 
In Algorithm 2, variable $\epsilon$ helps to define how far $z_{op,K}$ has to be from the 
low and up before the binary search terminates. BSWC outputs sets $M$ and 
$A$ when $z_{op,K}$ reaches its maximum possible value which is $z_{op,K} = up$ (line 
43).

The concept of BSWC is illustrated in Figure 4.2. Node-1 is the sink, 
and the distance between nodes is 2 m. The communication range of sensor 
nodes is 3.5 m. Sensor nodes generate one data packet every 100 seconds and 
the required energy for sending and receiving a data packet is 0.05 and 0.06 
J per packet respectively. Consequently, the energy consumption rate per 
second for sensor node-2 to 9 is $R = \{0.0082, 0.0071, 0.006, 0.0049, 0.0038,$
$0.0027, 0.0016, 0.0005\}$ respectively. The initial energy of all sensor nodes is 
54 J, meaning that the lifetime of sensor node-2 to 9 before the first charging 
round starts is $\{1.8, 2.1, 2.5, 3, 3.9, 5.5, 9.3, 30\}$ hours respectively and the set
Algorithm 2 Binary Search Wireless Charging (BSWC) algorithm.

1: Input $G(V,E), C_p = \{z_1, z_2, z_3, \cdots, z_n\}, K$
2: Output $M = \{n_0, n_1, n_2, \cdots, n_K, n_0\}$, $A = \{t_{1,p}, t_{2,p}, t_{3,p}, \cdots, t_{K,p}\}$
3: $M = NULL; A = NULL; z_{op,K} = 0; find = false$
4: while $K > 0$ and !find do
5:   $M = M \cup n_0$
6:   if $k == N$ then
7:     $z_{op,K} = Z_{max}$
8:   else
9:     if $K == 1$ then
10:        $z_{op,K} = z_{K,p-1}$
11:     else
12:        $z_{op,K} = Z_{K+1,p-1}$
13:     end if
14:   end if
15:   for $i \leftarrow 1$ to $K$ do
16:     $M = M \cup n_i; t_{i,p} = 0$
17:   end for
18:   $dis_p = TSP(M); b_m = B_m - (dis_p \times e_m)$
19:   if $b_m > 0$ then
20:     $finish = false; temp = 0; up = z_{N+1}; low = z_{op,K}$
21:     while !finish do
22:        $Totalt = 0$
23:        for $i \leftarrow 1$ to $K$ do
24:           if $((z_{op,K} - z_{i,p-1}) \times r_i + e_{i,p-1}) \leq B_s$ then
25:              $t_{i,p} = ((z_{op,K} - z_{i,p-1}) \times r_i)/e_{rx}$
26:           else
27:              $t_{i,p} = (B_s - e_{i,p-1})/e_{rx}$
28:           end if
29:           $Totalt + = t_{i,p}; A = A \cup t_{i,p}$
30:        end for
31:     if $Totalt \times e_{tx} > b_m$ then
32:        if $z_{op,K} > (low + \epsilon)$ and find then
33:           $up = z_{op,K}; z_{op,K} = (up + low)/2$
34:        else
35:           $finish = true$
36:        end if

else
    find = true;
    if temp == Total then
        finish = true;
    else
        if $z_{op,K} < (up - \epsilon)$ then
            temp = Total; low = $z_{op,K}$; $z_{op,K} = (up + low)/2$;
        else
            finish = true;
        end if
    end if
end if
end while
else
    find = false;
    if !find then
        $M = NULL$; $A = NULL$; $K -= 1$
    else
        break;
    end if
end if
end while
of sorted nodes is $C_1 = \{2, 3, 4, 5, 6, 7, 8, 9\}$. All other relevant parameters are listed in Table 4.2.

BSWC proceeds as follows:

Line 3: Sets $M$ and $A$ are initialized to NULL and variable find is set to false. The target lifetime $z_{op,K}$ is set to 0.

Line 4: Based on the network parameters which are listed in Table 4.2, the condition $K = 5 > 0$ is true and BSWC goes to line 5.

Lines 5-15: Node 1 is added to the tour $M = \{1\}$ and the target lifetime $z_{op,5} = 5.5$ hours, which is that of node 6, shown as $z_{6,0}$ in the sorted set $C_1$.

Lines 16-18: The first $K = 5$ nodes in $C_1$ are added to the tour to yield $M = \{1, 2, 3, 4, 5, 6, 1\}$ and their charging time is $A = \{0, 0, 0, 0, 0, 0\}$.

Line 19: BSWC calculates the shortest travel path for the six sensor nodes in $M$. Thus the $dis_1$ is equal to 20 m. Then, BSWC calculates the mobile
4. BINARY SEARCH WIRELESS CHARGING

charger’s residual energy; i.e., \( b_m = (1000 - (20 \times 40)) = 200J \).

Lines 20-21: The condition \( b_m = 200 > 0 \) is true, so BSWC goes to line 21 and initializes the following variables: \( \text{finish} \) and \( \text{temp} \). These variables are set to \( false \) and 0. BSWC also initializes the lower and upper bound of the binary search algorithm. The lower-bound is set to \( z_{6,0} \) or \( low = 5.5 \) hours. The upper-bound is set to \( z_{N+1} \), which is the lifetime of sensor node 9 when its battery is fully charged; i.e., \( up \) is equal to 55.5 hours.

Lines 24-31: BSWC calculates the charging time for the five sensor nodes in \( M \) to be charged up to \( z_{op,5} = 5.5 \) hours. The charging times for node 2 to 6 are \( A = \{38.3, 38.3, 38.3, 36.7, 18.2\} \) seconds and the total charging time is \( Total_t = 169.8 \) seconds.

Lines 32-38: The required energy to charge the five sensor nodes in \( M \) up to 5.5 hours is \( Total_t \times 3 \), which is equal to 509.4 J. Hence, the condition in line 32 is \( true \) and BSWC goes to line 33. The variable \( \text{find} \) remains \( false \), therefore, BSWC goes to line 35 and it sets the variable \( \text{finish} = true \).

Lines 53-54: At this point, \( \text{find} \) is \( false \), meaning the mobile charger’s energy is not sufficient to visit and charge sensor nodes in \( M \) up to \( z_{op,5} \). In this case, BSWC removes the nodes from \( M \) and reduces the number of visited sensor nodes to four \( K = 4 \), and restarts from line 4.

In the second iteration, where \( K = 4 \), \( M = \{1, 2, 3, 4, 5, 1\} \) and \( z_{op,4} = 3.9 \) hours, the shortest travel distance between \( |M| \) is \( dis_1 = 16 \) m and \( b_m = (1000 - (16 \times 40)) = 360J \). The charging times for nodes 2 to 5 are \( A = \{38.3, 38.3, 26.2, 14.4\} \) respectively. Hence the total charging time is \( Total_t = 117.6 \) seconds. The total required energy to charge the four sensor nodes in \( M \) up to a lifetime of 3.9 hours is \( Total_t3 = 352.8J \), which is less
than the remaining energy of the mobile charger \( b_m = 360 \text{J} \). Therefore, BSWC goes to line 38.

Lines 38-49: The variable \texttt{find} is set to \texttt{true} – line 39. The total charging time of sensor nodes has been changed in two consecutive iterations of the binary search (zero and 117.6 seconds) - line 40. This means that sensor nodes are not charged up to their maximum battery capacity yet and \( z_{op,4} \) can be increased further. Therefore, BSWC goes to lines 43 to 44 and finds a new target lifetime between 3.9 and 55.5, which is \( z_{op,4} = 29.7 \text{ hours} \). BSWC then goes to line 22 and calculates the new charging time for sensor nodes based on the new \( z_{op,4} \) value.

BSWC ends when the target lifetime is four hours, with a final node sequence of \( M = \{1, 2, 3, 4, 5, 1\} \). The corresponding charging times are \( A = \{38.3, 38.3, 27.2, 16.2\} \). The residual energy of the mobile charger upon arrival at the sink is zero.

\subsection{Analysis}

The time complexity of BSWC depends on how many times it calls the TSP solver to calculate the tour length - (line 19), and how many times it uses the binary search algorithm - (lines 22 to 50). The algorithm stops using the TSP solver when the target lifetime is achieved. Therefore, the worst case is when BSWC starts from \( K \) and the target cannot be achieved until \( K = 1 \). This means the proposed algorithm uses the TSP solver for a maximum of \( K \) times. After the target lifetime is achieved, BSWC uses binary search to find the optimal target lifetime. If \( \epsilon \) is considered 1, the worst case is when the optimal target lifetime is \( z_{op,K} \cong 0 \) and the binary search
iterates for \(\log_2(z_{N+1})\) times. Hence, the time complexity of the algorithm is \(O(K \times O(TSP) + O(\log_2(z_{N+1})))\). If a TSP solver that uses Christofides’ heuristic [126] is used, which has a time complexity of \(O(n^3)\), the resulting time complexity is \(O(K^4 + O(\log_2(z_{N+1})))\). Consequently, the time complexity of BSWC is \(O(K^4)\). This is in comparison to GP, which has a time complexity of \(O(K!K^2)\).

In this stage, the correctness of BSWC is shown.

**Theorem 3.** Assume BSWC and the optimal charging algorithm use the same TSP solver with maximum achievable lifetime set to \(Z_{\text{max}}\) over \(N\) sensor nodes. Let \(Z_{BSWC}\) denote the network lifetime when using BSWC. If \(K = N\), then BSWC guarantees that \(Z_{BSWC}\) is equal to \(Z_{\text{max}}\).

**Proof.** In Equ 4.8, the optimal charging algorithm finds the shortest path with length \(\text{dis}_N\) for the mobile charger to visit \(N\) sensor nodes. Hence, the residual energy \(b_{m,N}\) of the mobile charger’s battery that is available for charging sensor nodes is,

\[
b_{m,N} = B_m - (\text{dis}_N \times e_{mv}) \tag{4.17}
\]

BSWC also finds the shortest path with length \(\text{dis}_K\) to visit \(K\) nodes (line 19). Similarly, for BSWC \(b_{m,K}\) is,

\[
b_{m,K} = B_m - (\text{dis}_K \times e_{mv}) \tag{4.18}
\]

When \(K = N\), because both algorithms use the same TSP solver, \(\text{dis}_N\) is equal to \(\text{dis}_k\), meaning \(b_{m,N} = b_{m,K}\). BSWC considers \(Z_{\text{max}}\) as the target
lifetime \( z_{op,K} = Z_{\text{max}} \) when \( K = N \) (line 6). The charging time for each sensor node \( i \) in the optimal charging algorithm is calculated based on Equ. 4.10. Dividing \( e_{i,p-1} \) by \( r_i \) and multiplying by \( r_i \), the \( t_{i,p} \) is equal to,

\[
t_{i,p} = \frac{(r_i \times \frac{B_{s}}{r_{\text{max}}}) - (\frac{e_{i,p-1}}{r_i} \times r_i)}{e_{rx}}
\]  (4.19)

Replacing \( B_{s}/r_{\text{max}} \) with \( Z_{\text{max}} \) and \( e_{i,p-1}/r_i \) with \( z_{i,p-1} \) in Equ. 4.19 the \( t_{i,p} \) is,

\[
t_{i,p} = \frac{r_i \times (Z_{\text{max}} - z_{i,p-1})}{e_{rx}}
\]  (4.20)

The calculated charging time for sensor node \( i \) in Equ 4.20 is the same as what BSWC uses to calculate the charging time for sensor node \( i \) in line 26 of Algorithm 2 where \( z_{op,K} = Z_{\text{max}} \). Therefore, BSWC also achieves \( Z_{\text{max}} \) as the target lifetime \( Z_{\text{BSWC}} = Z_{\text{max}} \).

**Theorem 4.** BSWC always finds a charging sequence with at least one sensor nodes if there is a possible charging sequence to increase the network lifetime.

**Proof.** Consider a sorted set of sensor nodes \( C_p = \{n_1, n_2, n_3, \cdots, n_N\} \), where \( z_{i,p-1} \leq z_{i+1,p-1} \) and the network lifetime is \( z_{1,p-1} \). This means node \( n_1 \) is charged and included in the possible charging sequence as that tour increases the network lifetime. Suppose BSWC returns no charging sequence. This only happens when the mobile charger has insufficient energy to even visit sensor node \( n_1 \) (lines 6 to 15 of Algorithm 2). This contradicts the fact that there is at least one possible charging tour in the network.

**Theorem 5.** In BSWC, the residual energy \( R_{m,K} \) of the mobile charger after round \( p \) satisfies \( R_{m,K} \geq 0 \).
Proof. In a round $p > 1$ with charging sequence $M$, the residual energy of the mobile charger’s battery to charge $K$ sensor nodes is $b_{m,K}$. The amount of energy spent charging sensor node $i$ in $M$ is $t_{i,p} \times e_{tx}$. Hence $R_{m,K}$, the residual energy of the mobile charger battery when it returns to the sink position is equal to,

$$R_{m,K} = b_{m,K} - e_{tx} \sum_{i=1}^{K} t_{i,p}$$

(4.21)

where $e_{tx} \sum_{i=1}^{K} t_{i,p}$ is the energy spent charging $K$ sensor nodes in round $p$.

Induction is used to prove this theorem. In a situation that there is no sensor node in charging sequence $M$, the travelled distance is zero $dis_0 = 0$. From Equ. 3.14, when $dis_0 = 0$, then $b_{m,K}$ is equal to $B_m$. On the other hand, when $K = 0$, then $\sum_{i=1}^{K} t_{i,p} = 0$. According to Equ. 4.21, when $b_{m,K} = B_m$ and $\sum_{i=1}^{K} t_{i,p} = 0$, then $R_{m,K} = b_{m,K} = B_m > 0$. Suppose $R_{m,j} \geq 0$ is satisfied for $0 < j < K$. According to line 19 of Algorithm 2, for the $(j + 1)$th sensor node to be charged, $b_{m,j+1}$ and $R_{m,j+1}$ are,

$$b_{m,j+1} = B_m - (dis_{j+1} \times e_{mv})$$

(4.22)

$$R_{m,j+1} = b_{m,j+1} - e_{tx} \sum_{i=1}^{j+1} t_{i,p}$$

(4.23)

If $b_{m,j+1} > 0$ holds in line 20 and also $b_{m,j+1} - e_{tx} \sum_{i=1}^{j+1} t_{i,p}$ holds in line 38, then $R_{m,j+1} \geq 0$ is satisfied. Otherwise, the mobile charger is unable to use its residual energy $b_{m,j+1}$ to charge the $(j + 1)$th sensor node, and it will be removed from the charging sequence in line 54. Thus, $b_{m,j+1}$ is,

$$b_{m,j+1} = B_m - ((dis_{j+1} - (dis_{j+1} - dis_j)) \times e_{mv}) = b_{m,j}$$

(4.24)
Consequently, the mobile charger will travel back to the sink position with $R_{m,j+1} \geq 0$.

**Theorem 6.** In BSWC with charging sequence $M$ and $K > 0$ sensor nodes, the maximum residual energy in the mobile charger’s battery after it finishes the charging round $p > 1$ is

$$R_{m,K} = b_{m,K} - \frac{T_{p-1} \sum_{i=1}^{K} r_i}{e_{rx}}.$$

*Proof.* From Equ. 4.21, $R_{m,K}$ is maximized if $e_{tx} \sum_{i=1}^{K} t_{i,p}$ is minimized, where $b_{m,K} > 0$ and $e_{tx} \sum_{i=1}^{K} t_{i,p} > 0$. Hence, it is required to minimize the charging time $t_{i,p}$ for each sensor node $1 \leq i \leq K$ in order to obtain the minimum value for $e_{tx} \sum_{i=1}^{K} t_{i,p}$.

In line 25 of Algorithm 2, when the amount of energy given to sensor node $i$ is larger than $B_s$, the charging time for sensor node $i$ is (line 28),

$$t_{i,p} = \left( B_s - e_{i,p-1} \right) / e_{rx} \quad (4.25)$$

The charging time of sensor node $i$ in round $p$ is minimum if it is charged up to $B_s$ in charging round $p - 1$. In this case, the residual energy of sensor node $i$ at the end of charging round $p - 1$ from Equ. 4.3 is,

$$e_{i,p-1} = B_s - T_{p-1} \times r_i \quad (4.26)$$

The $e_{i,p-1}$ in Equ. 4.25 is replaced with the one from Equ. 4.26 then $t_{i,p}$ is,

$$t_{i,p} = \left( T_{p-1} \times r_i \right) / e_{rx} \quad (4.27)$$

Based on Equ. 4.27, the total required energy to charge all $K$ sensor nodes in
charging round $p$ is $\frac{r_p \sum_{i=1}^{K} r_i}{c_{rx}}$. The maximum residual energy of the mobile charger when it goes back to the sink node position is

$$R_{m,K} = b_{m,K} - \frac{T_{p-1} \sum_{i=1}^{K} r_i}{c_{rx}} \quad (4.28)$$

**Theorem 7.** Let $R_{m,K} = R_{m,K-1} = 0$ for $K > 2$. Also, let $\text{Lifetime}_K$ denote the maximum achievable network lifetime when the mobile charger charges $K$ sensor nodes and $\text{Total}_K$ be the total charging time for $K$ nodes. If $z_{K,p-1} < \text{Lifetime}_K < z_{K+1,p-1}$ then BSWC finds a charging sequence with $K - 1$ sensor nodes and achieves $\text{Lifetime}_{K-1}$ where $\text{Total}_{K-1} > \text{Total}_K$ and $\text{Lifetime}_{K-1} \leq \text{Lifetime}_K$.

**Proof.** If $K$ sensor nodes are charged up to $\text{Lifetime}_K$ in charging round $p$ where $z_{K,p-1} < \text{Lifetime}_K < z_{K+1,p-1}$, the travelled distance is $\text{dis}_K$. On the other hand, if the number of nodes reduced to $K - 1$ in charging round $p$ and are charged up to $\text{Lifetime}_{K-1} \geq z_{K,p-1}$, then the travelled distance $\text{dis}_{K-1}$ is,

$$\text{dis}_{K-1} = \text{dis}_K - d_K \quad (4.29)$$

where $d_K$ is the additional tour length when visiting sensor node $K$. From Equ. 4.29, it is concluded that $\text{dis}_K \geq \text{dis}_{K-1}$. On the other hand, based on Equ. 4.23 where $R_{m,K} = R_{m,K-1} = 0$,

$$b_{m,K} = \text{Total}_K \quad (4.30)$$

$$b_{m,K-1} = \text{Total}_{K-1} \quad (4.31)$$
From equations 4.18, 4.30 and 4.31, when $\text{dis}_K \geq \text{dis}_{K-1}$ it can be concluded that $\text{Total}_{K-1} > \text{Total}_K$.

Theorem 7 is significant because the GP algorithm charges $K$ sensor nodes up to $z_{op,p}$, $z_{K,p-1} < z_{op,p} < z_{K+1,p-1}$ when $z_{K+1,p-1}$ is not achievable. Theorem 7 shows that BSWC uses the mobile charger’s energy for charging sensor nodes rather than spending it on travel.

To illustrate this key advantage, consider Figure 4.3. A sorted set $C_p = \{n_1, n_2, n_3\}$ is calculated where the current lifetime of the corresponding nodes are $z_{1,p-1} = 10$ hours, $z_{2,p-1} = 10$ hours and $z_{3,p-1} = 30$ hours respectively. The energy consumption rate of each node is $1mJ/s$. The other network parameters are those outlined in Table 4.2. Base on the charging model of the GP algorithm, in charging round $p$, $z_{op,p} = 12$ hours is considered as the best target lifetime, and the mobile charger charges node $n_1$ and $n_2$. At the beginning of charging round $p+1$, the nodes’ lifetime is $z_{1,p} = 11$, $z_{2,p} = 11$ and $z_{3,p} = 29$ hours by considering $t_c = 1$ hour. In charging round $p+1$ where $C_p = \{n_1, n_2, n_3\}$, GP finds $z_{op,p} = 13$ hours as the target lifetime for charging sequence $\{n_1, n_2\}$. Hence, at the end of charging round $p+1$, the nodes lifetime is $z_{1,p+1} = 12$, $z_{2,p+1} = 12$ and $z_{3,p+1} = 28$ hours and the network lifetime is 12 hours.

Now if BSWC is used in charging round $p$, because the target lifetime $z_{3,p-1} = 30$ hours is not achievable for charging sequence $\{n_1, n_2, n_3\}$, the mobile charger charges sensor node $n_1$ only. Hence, the nodes’ lifetime before charging round $p+1$ starts is $z_{1,p} = 22$, $z_{2,p} = 9$ and $z_{3,p} = 29$ hours. In
charging round \( p + 1 \) where \( C_p = \{n_2, n_1, n_3\} \), the mobile charger simply charges sensor node \( n_2 \) and the nodes lifetime at the end of charging round \( p+1 \) is \( z_{1,p+1} = 21 \), \( z_{2,p+1} = 12 \) and \( z_{3,p+1} = 28 \) hours. The network lifetime is 12 hours but the total charging time in rounds \( p \) and \( p+1 \) is 17 hours for BSWC while for GP algorithm only obtains 8 hours.

### 4.4 Evaluation - BSWC

This section presents an evaluation of BSWC. The first task is to determine parameter \( K \). This parameter is important because if there are multiple charging rounds, the sensor nodes can run forever if at the end of each charging round \( p \), their lifetime is larger than \( T_{p+1} \). The maximum tour length \( T_{max} \) incurred to visit these \( K \) nodes and charging them up to \( B_s \) is,

\[
T_{max} = \frac{K \times B_s}{\eta \times \epsilon_{tx}} + \frac{(K + 1) \times d_{max}}{v} + t_c
\]  

(4.32)

where \( \frac{K \times B_s}{\eta \times \epsilon_{tx}} \) is the required time to charge \( K \) sensor nodes up to \( B_s \) and \( \frac{(K+1) \times d_{max}}{v} \) is the maximum travel time to visit \( K \) nodes. The variable \( d_{max} \) is the maximum distance between two nodes.
Based on Equ. 4.32, when $K$ is increased, $B_m$ needs to increase proportionally as well to ensure there is sufficient energy to charge $K$ nodes. The value of $B_m$, which is the maximum required energy to visit and charge $K$ sensor nodes up to $B_s$, is calculated as follows,

$$B_m = \frac{(K \times B_s)}{\eta} + ((K + 1) \times d_{max} \times e_{mv})$$

(4.33)

where $\frac{(K \times B_s)}{\eta}$ is the required energy to change $K$ sensor nodes up to $B_s$ and $((K + 1) \times d_{max} \times e_{mv})$ is the maximum required energy for the mobile charger to travel between $K + 1$ sensor nodes.

The sufficient condition to ensure a node remains alive forever is that $B_s$ must be large enough such that the node with $r_{max}$ consumption rate and maximum battery capacity stays alive for $T_{max}$ time. The largest possible $r_{max}$ results when one sensor node forwards $N - 1$ data packets to the sink node. Hence, for each network size, Equ. 4.34 calculates $B_s$, which is the maximum battery capacity that guarantees no sensor nodes die during $T_{max}$ time if their batteries are full:

$$B_s = ((N - 1) \times r_{rx} + N \times r_{tx}) \times p_r \times T_{max}$$

(4.34)

where $r_{rx}$ is the sensor node energy consumption for receiving a data packet from its one-hop neighbor and $r_{tx}$ is the sensor node energy consumption for sending a data packet to its one-hop neighbor. The variable $p_r$ is the data packet generation rate of sensor nodes per second, which is fixed for all nodes.
4. BINARY SEARCH WIRELESS CHARGING

Tab. 4.3: Simulation parameters.

<table>
<thead>
<tr>
<th>Simulation parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network area</td>
<td>200 x 200 m</td>
</tr>
<tr>
<td>Sensor’s communication range</td>
<td>2m</td>
</tr>
<tr>
<td>Initial sensor’s battery energy</td>
<td>75% of $B_s$</td>
</tr>
<tr>
<td>$p_t$</td>
<td>0.01 packet per second</td>
</tr>
<tr>
<td>$c_{tx}$</td>
<td>3 W</td>
</tr>
<tr>
<td>$c_{mv}$</td>
<td>40 Watt/meter</td>
</tr>
<tr>
<td>$v$</td>
<td>1 m/s</td>
</tr>
<tr>
<td>$r_{tx}$</td>
<td>0.05 J for a packet</td>
</tr>
<tr>
<td>$r_{rx}$</td>
<td>0.06 J for a packet</td>
</tr>
<tr>
<td>$\eta$</td>
<td>40%</td>
</tr>
<tr>
<td>$t_c$</td>
<td>1 hour</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>1</td>
</tr>
</tbody>
</table>

A series of networks with 20, 30, 40 and 50 sensor nodes is considered in the study. The nodes are randomly distributed in network area of 200 x 200 m while the network is connected. For each network, the simulation runs for $K = 1$ to minimum value of $K$ that ensures all sensor nodes remain alive perpetually. Brute-force is used to calculate the length of the tour for small values of $K$. Table 4.3 shows other simulation parameters.

4.4.1 Results

Figures 4.4 and 4.5 show $d_{max}$ and calculated $B_m$ for each network size. The $N_{20}$, $N_{30}$, $N_{40}$ and $N_{50}$ are used to denote networks with 20, 30, 40 and 50 sensor nodes respectively. As shown in Figure 4.4, the $d_{max}$ value for $N_{30}$ is larger than $N_{40}$. As per Equ 4.33, a larger $d_{max}$ value means higher $B_m$ value. Hence, from Figure 4.5, the $B_m$ value for $N_{30}$ is 10% larger than $N_{40}$. This means if the mobile charger has a fixed battery size, then it will charge fewer sensor nodes in $N_{30}$ as compared to $N_{40}$ in each charging round. Therefore, the travelled distance of the mobile charger has a direct impact
on its charging time. This factor is ignored in works such as [61].

Figure 4.6 shows the achieved network lifetime for different network sizes when $K$ starts from one and increases by one iteratively until a $K$ is found
that results in perpetual operation. The minimum $K$ value for $N_{20}$, $N_{30}$, $N_{40}$ and $N_{50}$ to achieve infinite lifetime is 9, 8, 8 and 7 respectively. Figure 4.6 demonstrates that for the same value of $K$, when the travelled distance of the mobile charger is longer, the achieved network lifetime is lower. In $N_{30}$, because its $d_{\text{max}}$ is larger than other networks, for the same value of $K$, the network lifetime is lower than other network sizes.

Next, a fixed $B_s = 100J$ and $B_m = 1000J$ values are considered for sensor nodes and mobile charger, and the best $K$ for BSWC is found to start in order to maximize network lifetime. The network topology is developed, with 15 nodes, used in [60] and depicted in Figure 4.7.

Figure 4.8 shows the lifetime for the network illustrated in Figure 4.7. The number of visited nodes changes from one to ten. With constant values of $B_s$ and $B_m$, a maximum network lifetime of 34.9 hours is achieved when the mobile charger visits three sensor nodes in each charging round. The network
lifetime increases from $K = 1$ to $K = 3$ and then decreases until it becomes 11.3 hours for $K > 8$. This is because when $B_m$ is fixed and the number of visited nodes increases, the travelled distance of the mobile charger also increases. Therefore, the mobile charger spends less energy charging sensor nodes. For example, the average travelled distance and charging time of the mobile charger in each charging round when $K = 3$ is 12.5 m and 121.91 seconds, while for $K = 5$, we have 16.5 m and 107.66 seconds respectively. For $K > 8$, the mobile charger does not have sufficient energy to charge and visit all $K$ nodes, meaning BSWC reduces $K$ until the target lifetime is achieved. Therefore increasing $K$ to more than eight does not have any effect on the network lifetime because of the mobile charger’s battery capacity constraint.

In the last simulation scenario, the performance of GP [60] and BSWC are compared in networks, as shown in Figures 4.2 and 4.7, with 9 and 15 nodes.
4. BINARY SEARCH WIRELESS CHARGING

Fig. 4.8: Achieved lifetime of the network shown in Figure 4.7 for $K = 1$ to $K = 10$.

$K = 5$ has been used for both network and $B_s = 100J$ and $B_m = 1000J$.

Figure 4.9 shows the achieved lifetime of the network in Figure 4.2 for GP and BSWC. The achieved lifetime for GP is seven hours and for BSWC is 8.4 hours, which is 20% more than GP. This is because in GP, when the target lifetime is achieved, the mobile charger returns to the depot even when there is some energy left in its battery while the visited sensor nodes are not fully charged. For BSWC, the target lifetime increases if there is leftover energy in the mobile charger’s battery (line 38 of Algorithm 2). Specifically, the observed left over energy of the mobile charger battery for GP is 130 J while for BSWC it is zero Joules. In addition, since GP does not consider the shortest travelling path, the mobile charger travels longer distance as compared to BSWC.

Figure 4.9 also shows the achieved lifetime of the network of Figure 4.7 for GP and BSWC. The achieved lifetime for GP is 6.9 hours, and for BSWC
28 hours, which is 400% more than GP. This is because in GP, when the network size increases, the mobile charger travels longer distance than when using BSWC. As a comparison, we observe that in the first seven charging rounds using GP, the mobile charger travels 116 m, while in BSWC, the mobile charger only travels 104 m. This equates to a saving of 480 J.

4.5 Conclusion

The lifetime of battery powered sensor nodes is a major performance bottleneck for WSNs. To this end, a key hypothesis is to investigate the viability of using a mobile charger to increase the lifetime of WSNs. The problem at hand can be formulated as an ILP and is shown to be NP-hard. As a solution, BSWC, a wireless charging algorithm that determines the movement and charging sequence of a mobile charger is proposed. BSWC minimizes
the travel distance of the mobile charger, and maximizes the charging time of sensor nodes. It has been shown that by selecting the right value for the mobile chargers battery size given the number of sensors to be visited, BSWC is able to guarantee all sensor nodes remain alive perpetually. The simulation results indicate that BSWC increases network lifetime by 400% as compared to Greedy-Plus.
Chapter 5

Conclusions

5.1 Introduction

The work reported in this thesis represents a multi-faceted study of WSNs and its extension, WSANs. The hypothesis and objectives were established after an extensive review of the literature. This review, in the first stage was broad with the goal of identifying research gaps in past studies that seek to address coordination issues in WSNs / WSANs. It then focused on two mobility based approaches. Firstly, the use of a mobile sink to mitigate energy holes in WSNs / WSANs by balancing the energy consumption rate of sensor nodes, and secondly, the use of a mobile charger to recharge the battery of sensor nodes. This led to novel observations, problems and corresponding solutions involving the use of a mobile sink / charger; the key topics addressed in this thesis.

In the following sections, the uniqueness and major findings of the literature review are first highlighted. This is then followed by an analysis of the strengths and constraints of the developed solutions.
5. CONCLUSIONS

5.2 State of the Art in WSNs / WSANs

The literature review presented in Chapter 2 is unique in both depth and breadth as compared to existing surveys. On one hand, the literature review represents the latest development in coordination. In particular, the introduction of actuators increases the complexity of conventional WSNs and poses new challenges. In such systems, sensor and actuator nodes must work hand-in-hand to collect and forward data, and act on sensed data collaboratively, promptly and reliably. In this respect, Chapter 2 highlights the importance of coordination between sensor and actuator nodes. The rest of the literature review was an in-depth review of problems related to mitigating energy holes in WSNs / WSANs using mobility based methods, more specifically, the use of mobile sinks and mobile chargers.

The literature on the application of mobile sinks proposed a combination of single-hop and multi-hop data forwarding patterns in alleviating energy holes as well as meeting sensed data delivery time. It also gave rise to the observation that visiting dense parts of a WSN / WSAN and giving priority to sensor nodes therein is an effective strategy. These findings were exploited by the algorithm reported in Chapter 3. Another observation was that the methods proposed for mobile chargers did not take into account the shortest travel path. Hence, saving the energy dissipated during travel increases the ability of the mobile charger to charge more sensor nodes. The method developed in the study as reported in Chapter 4 capitalized on this intuitive hypothesis.
5.3 Effectiveness of Mobile Sink

In order to alleviate the energy holes problem, a mobile sink can collect sensed data directly from sensor nodes, and thereby, help sensor nodes save energy that otherwise would be consumed in multi-hop communications. Considering the required delivery time of sensed data, visiting all sensor nodes may exceed the required delivery time of collected data packets. In WSNs with a large number of sensor nodes and limited delivery time for data packets, the time to visit all sensor nodes may exceed the required packet delivery time. Therefore, in order to meet packets delivery time, a mobile sink was programmed to visit only a set of sensor nodes called Rendezvous Points (RPs). Each RP is similar to a static sink, where sensor nodes forward their sensed data to the closest RP which is then collected by a mobile sink. Selecting a set of RPs such that the travel time between them does not exceed the required data packet delay and also uniformly balances sensor nodes energy consumption rates is a well-known mobile sink path selection problem.

As shown in Chapter 3, the proposed method addressed the major shortcoming of past solutions. In particular, CB [34] does not select two RPs from the same cluster, causing long data forwarding paths from sensor nodes to RPs and non-uniform energy depletion. RD-VT [35] traverses the SMT in pre-order to select the set of RPs that in turn results in long data forwarding paths to sensor nodes located in different parts of the SMT. RP-UG [36] considers a utility function that is dependent on physical distance. However, due to existence of obstacles, physical distance is not a reliable indicator of energy consumption. In addition, the time complexity of RP-UG grows
5. CONCLUSIONS

exponentially whenever its $L_0$ parameter value is reduced.

Chapter 3 formulated the mobile sink path selection problem for delay tolerance applications and showed its NP-hardness. Then, a heuristic algorithm called WRP, was presented to control the movement of a mobile sink in WSNs. WRP aimed to minimize and balance energy expenditure uniformly amongst sensor nodes in order to prevent the formation of energy holes whilst ensuring sensed data were collected on time. The results, obtained via computer simulation, indicated that WRP was a superior method as compared to previous methods and reduced network energy consumption 22% more than CB and distributed energy consumption of sensor nodes 39% and 44% better than CB and RD-VT, respectively.

5.4 Application of Mobile Wireless Charger

The next approach considered to mitigate energy holes was the use of a mobile charger along with rechargeable sensor nodes. In this method, a rover or an autonomous robot, equipped with a wireless charger, visits sensor nodes to replenish their battery up to a certain level. In the studied approach, the mobile charger starts its charging tour from the base station and returns to the same position after charging sensor nodes to recharge its battery. As the mobile charger has finite battery capacity, and considering the number of sensor nodes, charging all sensor nodes in one run may not be feasible. In this case, a mobile charger needs to select a set of sensor nodes to be charged in each charging tour. The work documented in Chapter 4 was focused on determining a set of sensor nodes that could be charged in each round and the
amount of energy given to them in order to maximize network lifetime. This was in contrast to other methods in the literature such as GP [60], J-RoC [61], ACTS [120] and NJNP [121], that did not consider the shortest travel path to visit sensor nodes. These methods also did not consider minimizing the residual energy of a mobile charger.

Chapter 4 formulated the mobile charger path selection problem as an ILP with the objective of maximizing network lifetime and showed its NP-hardness. Therefore, a heuristic method called BSWC was presented, whereby the mobile charger preferentially visited sensor nodes with the shortest lifetime. BSWC used the binary search algorithm to find the target lifetime in order to minimize the residual energy of the mobile chargers battery as well as using the shortest Hamiltonian path to minimize energy consumption due to travel between nodes. The validity of the algorithm was proven mathematically and a formula was derived to provide sufficient conditions that ensured infinite network lifetime. BSWC was validated via computer simulation. The results showed that it increased network lifetime by 400% as compared to GP.

5.5 Future Work

As a key future work, WRP can be extended to multiple mobile sinks / rovers case. This case, however, is non-trivial as it involves sub-problems such as interference and coordination between rovers. A large WSN can be partitioned into smaller areas where each area is assigned a mobile sink. Each mobile sink can be considered as an actuator node that collects data
from sensor nodes in its partition. A coordination platform among mobile sinks needs to be developed to transfer data collected by mobile sinks to the base station. The approach could be developed based on actuator-to-actuator coordination algorithms reviewed in Chapter 2. The required delivery time of data in each partition can be considered as task completion time. Therefore, WRP can still be run by each mobile sink to satisfy the task completion time of each partition.

Currently distributed or centralized coordination of multiple mobile sinks is still a challenge. As highlighted in Chapter 2, the centralized coordination models have high delays and as a consequence, they are not suitable for tasks that require fast completion time. On the other hand, distributed coordination model incurs a significant amount of communication overheads between mobile sinks. More work is required to address these issues and develop effective solution.

Finally, BSWC can be extended to use multiple mobile chargers as well. A possible approach is to divide the network area into smaller partitions based on the size of each mobile charger’s battery and deployment density of sensor nodes. A mobile charger can then be assigned to each partition.
Bibliography


[42] S. Soro and W. Heinzelman, “Prolonging the lifetime of wireless sensor networks via unequal clustering,” in Proceedings of 19th IEEE In-


[81] M. Vuong, H. Nguyen, H. Nguyen, and D. Dinh, “Wireless sensor actor networks and routing performance analysis,” in *Proceedings of Inter-
Bibliography 148


[106] M. Marta and M. Cardei, “Improved sensor network lifetime with mul-


Management Symposium (APNOMS), Seoul, Korea, September 2012, pp. 1–7.


