On Wireless Power Transfer and Max Flow in Rechargeable Wireless Sensor Networks

Tengjiao He
University of Wollongong

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:

This work is copyright. Apart from any use permitted under the Copyright Act 1968, no part of this work may be reproduced by any process, nor may any other exclusive right be exercised, without the permission of the author.

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.

Unless otherwise indicated, the views expressed in this thesis are those of the author and do not necessarily represent the views of the University of Wollongong.

Recommended Citation

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
On Wireless Power Transfer and Max Flow in Rechargeable Wireless Sensor Networks

A thesis submitted in partial fulfilment of the requirements for the award of the degree

Doctor of Philosophy

from

UNIVERSITY OF WOLLONGONG

by

Tengjiao He
Bachelor of Engineering (Computer Engineering)
School of Electrical, Computer and Telecommunications Engineering
August 2016
Statement of Originality

I, Tengjiao He, declare that this thesis, submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institutions.

Signed

Tengjiao He
August, 2016
Abstract

A typical Wireless Sensor Network (WSN) includes numerous sensor nodes that gather information from their environment. Sensor nodes then send sensed information to one or more sinks via one or more hops. A fundamental limitation is that sensor nodes have finite lifetime. Consequently, current research effort is focused on energy harvesting WSNs whereby sensor nodes harvest energy from their environment; examples of which include solar and wind. However, one key problem is that the energy supply from these environmental sources is not fixed. This thus motivates researchers to apply advances in Wireless Power Transfer (WPT) to power sensor nodes. To date, existing works on WPT aim to extend network lifetime via mobile or static wireless chargers. Those that use mobile chargers focus on trajectory planning. On the other hand, works on static chargers aim to place chargers strategically. In both cases, the goal is to maximize some objective.

One such objective is data gathering. This operation is critical because the amount of data determines the sensing quality of the area monitored by sensor nodes. Past works on data gathering have two main objectives: throughput maximization and fair rate allocation. The former means one or more sinks aim to extract the maximum data from a WSN. To date, in order to increase data gathering rate, past works have considered routing approaches or optimized the location of sinks. In addition, some works also consider deploying additional sensor nodes.
Unlike past works, this thesis first considers a novel problem that jointly considers WPT, throughput maximization, link scheduling, data routing and chargers deployment. Specifically, it aims to “upgrade” the recharging rate of a finite number of “bottleneck” nodes using so called Auxiliary Chargers (ACs) equipped with WPT capability. The problem, called ACP-MF, is modeled as a Mixed Integer Linear Program (MILP). As the problem is NP-hard, in order to solve large problem instances, this thesis outlines three novel solutions to place ACs: (i) Path, which preferentially upgrades nodes on the shortest path amongst paths from sources to sinks, (ii) Tabu, a meta-heuristic that first uses Path as the initial solution. It then searches for a neighboring solution that yields a higher max flow rate, and (iii) LagOP, which approximates the said MILP using Lagrangian and sub-gradient optimization. Evaluation results show that Tabu has the best performance; it is able to achieve 99.40% of the max flow rate derived by MILP in tested scenarios.

This thesis then aims to deploy ACs to maximize the minimum source rate. Specifically, in a rechargeable WSN, sensor nodes have varying energy harvesting rates. This impacts sensing quality because some sensor nodes or sources will transmit less data to one or more sinks. Consequently, it is important to upgrade nodes with low energy harvesting rates. This thesis proposes a novel problem called ACP-MM. Its aim is to place a finite number of ACs to maximize the minimum sensing rate in a WSN. The ACP-MM problem is formulated as a MILP. As the MILP is intractable for large-scale WSNs, this thesis devises two heuristic algorithms to place ACs: (i) GND, which checks all sensor nodes and parks an AC next to a node that result in the highest increase in max-min rate, and (ii) OUED, which first relaxes the MILP into a Linear Program (LP) and uses it to share one unit of energy among sensor nodes in each iteration and then upgrades the sensor node with the highest one-unit share. Evaluation results show that the max-min rate obtained by GND and OUED is respectively within 99.34% and 97.97% of the max-min rate derived by MILP in small networks.

The last contribution of this thesis is a novel problem that jointly considers data
routing, rate allocation, energy sharing and link scheduling in order to maximize the minimum sensing rate. In particular, it assumes that sensor nodes harvest energy from both solar and Radio Frequency (RF) signal from their neighbors. In other words, nodes with a high energy harvesting rate are able to share their energy with nodes that have a low energy harvesting rate. A novel problem called RFES-MM is proposed: determine the time used to share energy and data transmission such that the minimum sensing rate is maximized. The problem is formulated and solved as a LP. In addition, this thesis contains a novel heuristic algorithm called CHMM. It provides a fixed route for each source and checks whether a given sensing rate meets all constraints. It employs binary search to reduce/improve the sensing rate in order to achieve the maximum common sensing rate. Experiment results show that RF energy sharing among nodes improves the max-min rate by 13.98%. In addition, the average gap between CHMM and the LP is 4.99%.
Acknowledgments

First and foremost, I would like to express my special gratitude to my supervisor, Associate Professor Kwan-Wu Chin. He spent a lot of his valuable time on improving my academic writing and research skill. I thank him for his patience and enthusiasm throughout my PhD studies.

I like to thank my co-supervisor Dr Sieteng Soh for his helpful comments to improve my papers and thesis.

Special thanks to my office mates, Dr He Wang, Dr Charlie Yang, Dr Luyao Wang, Mr Tianle Liu, Mr Shichao Fu, Mr Sen Zhang, Mr Wei Han and Miss Heidi Xu, for their helpful discussions.

Finally, I would like to thank my family for their great love and encouragement throughout my life. Without their love, it is impossible for me to reach this milestone.
Contents

Abstract II

Acknowledgments V

Abbreviations XIII

1 Introduction 1

1.1 Background . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

1.2 Problem Space . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

1.2.1 Auxiliary chargers placement to maximize flow rate at the sink
(ACP-MF) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

1.2.2 Auxiliary Chargers placement to maximize the minimum sens-
ing rate (ACP-MM) . . . . . . . . . . . . . . . . . . . . . . . 12

1.2.3 RF energy sharing to maximize the minimum sensing rate
(RFES-MM) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

1.3 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

1.3.1 ACP-MF . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

1.3.2 ACP-MM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

1.3.3 RFES-MM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

1.4 Publications . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

VI
1.5 Thesis Structure .................................................. 17

2 Literature Review ........................................ 19
  2.1 Wireless Charging ............................................. 19
    2.1.1 Mobile Chargers .......................................... 21
      2.1.1.1 Single Charger ..................................... 21
      2.1.1.2 Multiple Chargers .................................. 32
    2.1.2 Static Chargers .......................................... 37
      2.1.2.1 One Hop Energy Transfer ......................... 37
      2.1.2.2 Multiple Hops Energy Transfer .................. 44
  2.2 Data Gathering ............................................. 49
    2.2.1 Throughput Maximization ............................ 49
    2.2.2 Fair Rate Allocation .................................. 54
  2.3 Summary .................................................. 61

3 ACs Placements and Max Flow ......................... 63
  3.1 Preliminaries ............................................. 64
  3.2 Problem Formulation ..................................... 65
  3.3 Proposed Solutions ...................................... 68
    3.3.1 Path .................................................. 69
    3.3.2 Tabu .................................................. 69
    3.3.3 LagOP ............................................... 72
  3.4 Evaluation ................................................ 75
    3.4.1 Results for Small Networks .......................... 77
    3.4.2 Results for Large Networks ....................... 81
  3.5 Conclusion ............................................. 87

4 ACs Placements and Min Flow Rate ................ 88
  4.1 Preliminaries ............................................. 89
  4.2 Problem Formulation ..................................... 90
## Contents

4.3 Proposed Solutions .................................................. 91  
  4.3.1 Greedy Node Deployment (GND) Algorithm .................. 92  
  4.3.2 One Unit Energy Deployment (OUED) Algorithm ............ 93  
4.4 Analysis ............................................................. 95  
4.5 Evaluation ............................................................ 100  
  4.5.1 Small Networks ................................................ 101  
  4.5.2 Large Networks ............................................... 106  
4.6 Conclusion ............................................................ 110  

5 Energy Transfer and Max Min Rate ................................. 112  
  5.1 Preliminaries ....................................................... 113  
  5.2 Problem Definition ............................................... 116  
  5.3 A Heuristic Solution ............................................. 118  
  5.4 Evaluation .......................................................... 122  
    5.4.1 Results ....................................................... 123  
  5.5 Conclusion .......................................................... 139  

6 Conclusion .............................................................. 142  

References ................................................................. 145  

Appendices ................................................................. 161
List of Figures

1.1 An example WSN. .......................................................... 1
1.2 Block diagram of a sensor node. ....................................... 3
1.3 An example WSN with two ACs and two sinks. .................. 12
1.4 An example WSN to show the relationship between energy consump-
tion and maximum data rate. ........................................... 13
1.5 A WSN example with energy and data transfer denoted by dotted
and solid arrows, respectively. ........................................... 14
2.1 Chapter 2’s structure ....................................................... 20
2.2 A MC re-fueling sensor nodes. ........................................... 22
2.3 A HMC re-fueling and collecting data from sensor nodes. ....... 22
2.4 Hexagonal cells covering a network. ................................. 25
2.5 An example charging tour. ............................................... 26
2.6 An example WSN in [1]. Areas separated by dotted circles represent
candidate stop locations of RFID readers. ........................... 28
2.7 (a) A charging path computed by the method in [2] that has the
minimum length, (b) A naive charging tour where the MC stops at
each node ................................................................. 29
2.8 Equilateral triangles covering a network. ........................... 38
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>An instance of ACP-MF.</td>
<td>68</td>
</tr>
<tr>
<td>3.2</td>
<td>Max flow with varying $\delta$ values.</td>
<td>78</td>
</tr>
<tr>
<td>3.3</td>
<td>Max flow with varying $</td>
<td>V</td>
</tr>
<tr>
<td>3.4</td>
<td>Max flow with varying $R^#$ values.</td>
<td>80</td>
</tr>
<tr>
<td>3.5</td>
<td>Max flow with varying $</td>
<td>S</td>
</tr>
<tr>
<td>3.6</td>
<td>Max flow with varying $R^#$ values in large networks.</td>
<td>83</td>
</tr>
<tr>
<td>3.7</td>
<td>Max flow with varying $\delta$ values in large networks.</td>
<td>84</td>
</tr>
<tr>
<td>3.8</td>
<td>Max flow with varying $</td>
<td>S</td>
</tr>
<tr>
<td>3.9</td>
<td>Impact of $</td>
<td>S</td>
</tr>
<tr>
<td>3.10</td>
<td>Jain’s fairness index of source rates.</td>
<td>87</td>
</tr>
<tr>
<td>4.1</td>
<td>Max-min rate with varying $</td>
<td>V</td>
</tr>
<tr>
<td>4.2</td>
<td>Max-min rate with varying $R^#$ values.</td>
<td>103</td>
</tr>
<tr>
<td>4.3</td>
<td>Max-min rate with varying $</td>
<td>S</td>
</tr>
<tr>
<td>4.4</td>
<td>Max-min rate with varying $\delta$ values.</td>
<td>106</td>
</tr>
<tr>
<td>4.5</td>
<td>Max-min rate with varying $R^#$ values in large networks.</td>
<td>107</td>
</tr>
<tr>
<td>4.6</td>
<td>Running time with varying $R^#$ values in large networks.</td>
<td>108</td>
</tr>
<tr>
<td>4.7</td>
<td>Max-min rate with varying $</td>
<td>S</td>
</tr>
<tr>
<td>4.8</td>
<td>Running time with varying $</td>
<td>S</td>
</tr>
<tr>
<td>4.9</td>
<td>Max-min rate with varying $\delta$ values in large networks.</td>
<td>110</td>
</tr>
<tr>
<td>4.10</td>
<td>Running time with varying $\delta$ values in large networks.</td>
<td>111</td>
</tr>
<tr>
<td>5.1</td>
<td>Max-min rate with varying $</td>
<td>S</td>
</tr>
<tr>
<td>5.2</td>
<td>The value of $R_E, T_E, R_E$ and $T_E$ with varying $</td>
<td>S</td>
</tr>
<tr>
<td>5.3</td>
<td>The value of $R_D, T_D, R_D$ and $T_D$ with varying $</td>
<td>S</td>
</tr>
<tr>
<td>5.4</td>
<td>Running time with varying $</td>
<td>S</td>
</tr>
<tr>
<td>5.5</td>
<td>Max-min rate with varying $P_T$.</td>
<td>128</td>
</tr>
<tr>
<td>5.6</td>
<td>The value of $R_E, T_E, R_E$ and $T_E$ with varying $P_T$.</td>
<td>129</td>
</tr>
<tr>
<td>5.7</td>
<td>The value of $R_D, T_D, R_D$ and $T_D$ with varying $P_T$.</td>
<td>130</td>
</tr>
<tr>
<td>5.8</td>
<td>Running time with varying $P_T$.</td>
<td>131</td>
</tr>
</tbody>
</table>
5.9 Max-min rate with varying $\eta$. 132
5.10 The value of $R_E, T_E, R_E$ and $T_E$ with varying $\eta$. 132
5.11 The value of $R_D, T_D, R_D$ and $T_D$ with varying $\eta$. 133
5.12 Running time with varying $\eta$. 134
5.13 Max-min rate with varying $|V|$. 135
5.14 The value of $R_E, T_E, R_E$ and $T_E$ with varying $|V|$. 136
5.15 The value of $R_D, T_D, R_D$ and $T_D$ with varying $|V|$. 136
5.16 Running time with varying $|V|$. 137
5.17 Max-min rate with varying $\mathcal{G}$. 138
5.18 The value of $R_E, T_E, R_E$ and $T_E$ with varying $\mathcal{G}$. 139
5.19 The value of $R_D, T_D, R_D$ and $T_D$ with varying $\mathcal{G}$. 140
5.20 Running time with varying $\mathcal{G}$. 140
List of Tables

1.1 Characteristics of example sensors [3] .......................... 3
1.2 Characteristics of some transceivers [4][3][5][6] .......................... 4
1.3 Characteristics of example MCUs [7][8][9][10][11][12] ...................... 4
1.4 Comparison of sensor node platforms [4][3] .......................... 6
1.5 Comparison of WPT technologies. ........................................ 10

2.1 A summary of past works on using single MC ........................ 31
2.2 A summary of past works that use multiple MCs. ..................... 36
2.3 A summary of past works on single hop energy transfer. ........... 43
2.4 A summary of past works on multiple hops energy transfer. ....... 50
2.5 A summary of works on throughput maximization .................... 55
2.6 A summary of [13][14] and [15]. ....................................... 60
2.7 A summary of works on LMM rate allocation ........................ 60

3.1 Notations and Definitions in ACP-MF problem ........................ 65
3.2 The value of parameters in the evaluation ............................ 76
3.3 A summary of results. .................................................. 86

5.1 Key Notations and Definitions in RFES-MM Problem ............... 115
5.2 Evaluation parameters .................................................. 122
### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>Auxiliary Charger</td>
</tr>
<tr>
<td>ACK</td>
<td>Acknowledgment</td>
</tr>
<tr>
<td>ACP-MF</td>
<td>Auxiliary Chargers placement to maximize the flow rate at the sink</td>
</tr>
<tr>
<td>ACP-MM</td>
<td>Auxiliary Chargers placement to maximize the minimum sensing rate</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>CH</td>
<td>Cluster Header</td>
</tr>
<tr>
<td>CHMM</td>
<td>Centralized Heuristic Max-min Rate</td>
</tr>
<tr>
<td>CTC</td>
<td>Clear to Charge</td>
</tr>
<tr>
<td>CTP</td>
<td>Collection Tree Protocol</td>
</tr>
<tr>
<td>DCLK</td>
<td>Distributed Coordination Local Knowledge</td>
</tr>
<tr>
<td>DVRP</td>
<td>Distance Constrained Vehicle Routing Problem</td>
</tr>
<tr>
<td>ET</td>
<td>Energy Transmitter</td>
</tr>
<tr>
<td>ETX</td>
<td>Expected Transmission Count</td>
</tr>
<tr>
<td>FCFS</td>
<td>First-Come-First-Serve</td>
</tr>
<tr>
<td>FIFO</td>
<td>First In First Out</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>GND</td>
<td>Greedy Node Deployment</td>
</tr>
<tr>
<td>GPS</td>
<td>Geographic Position System</td>
</tr>
<tr>
<td>H-AP</td>
<td>Hybrid Access Point</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>HMC</td>
<td>Hybrid Mobile Charger</td>
</tr>
<tr>
<td>ID</td>
<td>Identification</td>
</tr>
<tr>
<td>ILP</td>
<td>Integer Linear Program</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>KKT</td>
<td>Karush-Kuhn-Tucker</td>
</tr>
<tr>
<td>LD</td>
<td>Lagrangian Dual</td>
</tr>
<tr>
<td>LoS</td>
<td>Line-of-sight</td>
</tr>
<tr>
<td>LLBP</td>
<td>Lagrangian Lower Bound Program</td>
</tr>
<tr>
<td>LMM</td>
<td>Lexicographic Max-Min</td>
</tr>
<tr>
<td>LP</td>
<td>Linear Program</td>
</tr>
<tr>
<td>MBN</td>
<td>Mobile Backbone Node</td>
</tr>
<tr>
<td>MC</td>
<td>Mobile Charger</td>
</tr>
<tr>
<td>MCR</td>
<td>Maximum Common Rate</td>
</tr>
<tr>
<td>MIP</td>
<td>Mixed Integer Program</td>
</tr>
<tr>
<td>MILP</td>
<td>Mixed Integer Linear Program</td>
</tr>
<tr>
<td>MINLP</td>
<td>Mixed Integer Nonlinear Program</td>
</tr>
<tr>
<td>MSR</td>
<td>Maximum Single Rate</td>
</tr>
<tr>
<td>MTSPD</td>
<td>Multiple Traveling Salesmen Problem with Deadlines</td>
</tr>
<tr>
<td>NDN</td>
<td>Named Data Networking</td>
</tr>
<tr>
<td>NJNP</td>
<td>Nearest-Job-Next-with-Preemption</td>
</tr>
<tr>
<td>NLP</td>
<td>Nonlinear Program</td>
</tr>
<tr>
<td>OUED</td>
<td>One Unit Energy Deployment</td>
</tr>
<tr>
<td>PA</td>
<td>Parametric Analysis</td>
</tr>
<tr>
<td>PSO</td>
<td>Particle Swarm Optimization</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>rWSN</td>
<td>Wireless Rechargeable Sensor Network</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RFES-MM</td>
<td>RF energy sharing among nodes to maximize the minimum sensing rate</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio Frequency Identification</td>
</tr>
<tr>
<td>RLT</td>
<td>Reformulation and Linearization</td>
</tr>
<tr>
<td>RTC</td>
<td>Request to Charge</td>
</tr>
<tr>
<td>SES</td>
<td>Smallest Enclosing Space</td>
</tr>
<tr>
<td>Sun SPOT</td>
<td>Sun Small Programmable Object Technology</td>
</tr>
<tr>
<td>TSP</td>
<td>Traveling Salesman Problem</td>
</tr>
<tr>
<td>UAV</td>
<td>Unmanned Aerial Vehicles</td>
</tr>
<tr>
<td>WBAN</td>
<td>Wireless Body Area Network</td>
</tr>
<tr>
<td>WISP</td>
<td>Wireless Identification and Sensing Platform</td>
</tr>
<tr>
<td>WPT</td>
<td>Wireless Power Transfer</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

Wireless Sensor Networks (WSNs) may contain tens of thousands of nodes deployed in a large area [16]. Each sensor node senses its environment for information such as humidity, temperature and pressure [17]. It then transfers sensed data to one or more sinks wirelessly. These sinks then transfer collected data to a remote center via the Internet for further analysis. An example WSN is shown in Figure 1.1. In particular, seven sensor nodes are deployed to monitor a sensing area. When a sensor node collects raw data from its surroundings, it transfers sensed data to the sink via other sensor nodes; this is also called multi-hop communications.

Figure 1.1: An example WSN.
There are generally two types of WSN applications: monitoring and tracking [18]. Monitoring applications include those that monitor human health and the environment. An example of the former is wireless body area networks (WBANs) [19]. A WBAN contains numerous sensors that are deployed on or inside a human body. These sensors monitor a person’s heart rhythm or blood pressure [20]; alternatively, they can be used to monitor an infant’s sleep condition [21]. As for environment monitoring, an example application is reported in [22], where 16 sensor nodes are used to detect the eruption of an active volcano. Another example is to use a WSN to monitor the vibration of a bridge [23]. Tracking applications include sensors with a Geographic Position System (GPS) such that a remote center knows the location of each sensor. These sensor nodes can then be used to track wild life. For example, in the ZebraNet project [24], zebras wear a collar with sensors and a GPS unit. Through their collar, a herd of zebras forms a WSN. The collars then send the positions of zebras to a base station, possibly with the help of collars worn by other zebras. Biologists then use collected data to study the migration patterns of zebras.

Apart from that, WSNs are also anticipated to play a critical role in realizing the Internet of Things (IoT). Specifically, sensor nodes, and smart phones communicate with each other to accomplish some tasks [25]. Once a user requests a specific service, IoTs may query a WSN to gather information, analyze collected information and provide an answer to the user. One example is a smart home system [26], where numerous sensors are used to monitor a home’s temperature and humidity.

A typical sensor node contains four subsystems [17]: sensing, communication, processing and power supply; see Figure 1.2. The sensing subsystem is used to collect sensory information. Table 1.1 lists numerous sensors. In particular, the sensing time of Fastrax iTRAX03 is at least 20 times longer than that of other sensors. Further, its energy consumption rate reaches 95 mW. By contrast, Dallas DS620U is the sensor with the second highest energy consumption of 2.4 mW.
1.1. Background

![Block diagram of a sensor node.](image)

Table 1.1: Characteristics of example sensors [3]

<table>
<thead>
<tr>
<th>Sensed Data</th>
<th>Example Sensor</th>
<th>Accuracy</th>
<th>Sensing time</th>
<th>Energy Consumption (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceleration</td>
<td>VTI SCA3000</td>
<td>1%</td>
<td>10 ms</td>
<td>0.36</td>
</tr>
<tr>
<td>Position</td>
<td>Fastrax iTRAX03</td>
<td>1.0 m</td>
<td>4.0 s</td>
<td>95</td>
</tr>
<tr>
<td>Air pressure</td>
<td>VTI SCP1000</td>
<td>150 Pa</td>
<td>110 ms</td>
<td>0.075</td>
</tr>
<tr>
<td>Temperature</td>
<td>Dallas DS620U</td>
<td>0.5 C</td>
<td>200 ms</td>
<td>2.4</td>
</tr>
<tr>
<td>Humidity</td>
<td>Sensorion SHT15</td>
<td>2%</td>
<td>200 ms</td>
<td>0.95</td>
</tr>
</tbody>
</table>

The communication subsystem contains a transceiver and an antenna. Existing sensor nodes use a variety of communication technologies. For example, underwater WSNs use acoustic [18]. However, the main drawbacks include low data rates and high propagation delays. By contrast, radio frequency (RF) wave has higher data rates. For example, current sensor node platforms typically use the Texas Instrument (TI) CC2420 chipset [27], which implements the IEEE 802.15.4 standard [28]. Some transceivers are shown in Table 1.2. Note that Zeevo ZV4002 is a Bluetooth transceiver. In addition, CC2420 has the lowest output power; i.e., -24 dBm. Referring to Figure 1.2, the processing subsystem bridges sensing and communication subsystems. It processes digital signal from the sensory subsystem. Table 1.3 shows popular processors [29][4]. The fastest processor to date is the Atmel ARM920T.
Table 1.2: Characteristics of some transceivers [4][3][5][6]

<table>
<thead>
<tr>
<th>Transceiver</th>
<th>Data rate (kb/s)</th>
<th>Frequency band (MHz)</th>
<th>Minimum output power (dBm)</th>
<th>Power sensitivity (dBm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC2420</td>
<td>250</td>
<td>2400</td>
<td>-24</td>
<td>-95</td>
</tr>
<tr>
<td>CC2500</td>
<td>500</td>
<td>2400</td>
<td>-12</td>
<td>-104</td>
</tr>
<tr>
<td>CC1000</td>
<td>76.8</td>
<td>433-915</td>
<td>-20</td>
<td>-109</td>
</tr>
<tr>
<td>CC1101</td>
<td>500</td>
<td>433-915</td>
<td>-6</td>
<td>-116</td>
</tr>
<tr>
<td>Zeevo ZV4002</td>
<td>57.6-723.2</td>
<td>2400</td>
<td>+4</td>
<td>-82</td>
</tr>
<tr>
<td>AT86RF231</td>
<td>250</td>
<td>2400</td>
<td>-17</td>
<td>-101</td>
</tr>
</tbody>
</table>

Table 1.3: Characteristics of example MCUs [7][8][9][10][11][12]

<table>
<thead>
<tr>
<th>Processors</th>
<th>Bits</th>
<th>Frequency (MHz)</th>
<th>Flash (kb)</th>
<th>RAM (kb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atmel ATmega128L</td>
<td>8</td>
<td>8</td>
<td>128</td>
<td>4</td>
</tr>
<tr>
<td>TI MSP430F1611</td>
<td>16</td>
<td>8</td>
<td>48</td>
<td>10</td>
</tr>
<tr>
<td>ARM 32 bit Cortex M3</td>
<td>32</td>
<td>8-48</td>
<td>256-512</td>
<td>32-100</td>
</tr>
<tr>
<td>TI MSP430G2955</td>
<td>16</td>
<td>8/16</td>
<td>56</td>
<td>4</td>
</tr>
<tr>
<td>Atmel ARM920T</td>
<td>32</td>
<td>180</td>
<td>4096</td>
<td>512</td>
</tr>
<tr>
<td>Atmel ATmega103L</td>
<td>8</td>
<td>8</td>
<td>128</td>
<td>4</td>
</tr>
</tbody>
</table>

with an operating frequency of 180 MHz. Lastly, the power supply subsystem stores and provides energy to a sensor node. Based on the battery type, a sensor node’s power supply subsystem can be classified into two categories: non-rechargeable and rechargeable. In particular, compared to a rechargeable battery, a non-rechargeable battery is cheaper and has higher energy density; see [3]. For example, in [30], a zinc-air non-rechargeable battery has an energy density of 1150 Wh/L. As a comparison, a NiMH rechargeable battery has an energy density of 175 Wh/L. However, NiMH can be recharged 1000 times [16].

To date, there are numerous commercial sensor node platforms; see Table 1.4 for examples. In particular, the platform from Sun Small Programmable Object Technology (Sun SPOT) [29] is the most expensive, at $750, due to its recharging ability. Sun SPOT also has the fastest processor. However, compared to other platforms, it requires a high operating voltage, at 3.7V. Atmel ATMega128L is the most popular processor among all listed platforms. With regards to transceivers, the
most popular is CC2420 from TI as it requires the lowest energy to operate; see Table 1.2. Apart from that, these platforms typically use an AA cell battery. In addition, they are limited in size with space for only two AA batteries. Consequently, energy poses a fundamental constraint.

Energy constraint is a key problem in WSNs. This is because a sensor node has a tiny size; see Table 1.4. Thus, the sensor nodes in Table 1.4 are powered only by two AA batteries. Therefore, their lifetime is limited, e.g., as per [31], the lifetime of a Mica2 mote is only one year. When a sensor node exhausts its energy, its neighbors may lose their path to the sink. In the worst case, a WSN becomes partition and useless. Note that nodes can be equipped with a larger battery. This, however, increases their size and cost.

To prolong the lifetime of sensor nodes, researchers have proposed numerous methods. These methods are categorized into two areas: energy conservation and energy upgrade. Specifically, they aim to conserve energy and ensure sensor nodes use their available energy judiciously. In this respect, power management is critical. In particular, as the energy consumption of the communication subsystem is significantly higher than other subsystems [17], sensor nodes operate in two states: wake-up and sleep. Nodes in sleep mode turn off all their functions to conserve energy. Thus, a key research area is devising wake/sleep scheduling algorithms [37][38][39] to reduce energy consumption whilst meeting one or more objectives; e.g., delay. Apart from that, energy-aware routing protocols are also important to better utilize the energy of sensor nodes. An example routing protocol is [40]. As sensor nodes have finite battery capacity, researchers have also sought methods to upgrade sensor nodes. Specifically replacing sensor nodes with a depleted battery. One approach is to employ a robot that roams a field to replace dead sensor nodes with fully charged ones [41][42]. One drawback is that robots require the location of sensor nodes. In addition, a robot needs access to sensor nodes, meaning they cannot be used in inaccessible or hostile terrains.

To overcome the finite battery of sensor nodes, researchers are investigating the
Table 1.4: Comparison of sensor node platforms [4][3]

<table>
<thead>
<tr>
<th>Node platform</th>
<th>Cost (USD$)</th>
<th>Processor</th>
<th>Transceiver</th>
<th>Voltage (V)</th>
<th>Size (mm)</th>
<th>Battery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mica2 [31]</td>
<td>99</td>
<td>Atmel ATMega128L</td>
<td>CC1000</td>
<td>2.7-3.3</td>
<td>24.3×24.3</td>
<td>2× AA batteries</td>
</tr>
<tr>
<td>MicaZ [32]</td>
<td>99</td>
<td>Atmel ATMega128L</td>
<td>CC2420</td>
<td>2.7-3.3</td>
<td>58×32</td>
<td>2× AA batteries</td>
</tr>
<tr>
<td>TelosB [33]</td>
<td>139</td>
<td>TI MSP430F1611</td>
<td>CC2420</td>
<td>2.1-3.6</td>
<td>66×32</td>
<td>2× AA batteries</td>
</tr>
<tr>
<td>.NOW [34]</td>
<td>125</td>
<td>ARM Cortex M3</td>
<td>AT86RF231</td>
<td>3.3</td>
<td>76 × 38</td>
<td>A NiCd battery</td>
</tr>
<tr>
<td>WiSense [35]</td>
<td>18.80</td>
<td>TI MSP430G2955</td>
<td>CC1101</td>
<td>1.8-3.6</td>
<td>42 × 42</td>
<td>A lithium coin cell</td>
</tr>
<tr>
<td>BTnode rev3</td>
<td>200</td>
<td>Atmel ATmega128L</td>
<td>Zeevo ZV4002</td>
<td>2-3</td>
<td>58.15 × 32.5</td>
<td>2× AA batteries</td>
</tr>
<tr>
<td>Sun SPOT [4]</td>
<td>750</td>
<td>Atmel ARM920T</td>
<td>CC2420</td>
<td>3.7</td>
<td>38 × 64</td>
<td>A Lithium-ion battery</td>
</tr>
</tbody>
</table>
use of energy harvesting technologies. The advances in energy harvesting technologies allow sensor nodes to collect ambient energy and convert them to electrical energy. Based on the energy source, these technologies are divided into two categories [16][43]: ambient and human energy source. The former includes solar, wind, heat and RF signal. Specifically, solar is the most popular ambient energy source; example sensor nodes that use solar include SunSPOT, Everlast and AmbimMax [16]. As per [44][45] and [46], the harvested energy of a solar cell is approximately 100 $mW/cm^2$ with a conversion efficiency of 15% during daytime. However, the power density of solar reduces to zero at night. Further, the amount of harvested energy from solar is often unpredictable and uncontrollable [4]. This is because solar energy is restricted by location, time of day and atmospheric condition. Wind power is also unpredictable. In [47], a wind energy collection system is shown to harvest 1200 mWh/day. However, a drawback is its large size. Compared to solar, thermal energy source is available continuously. In [43], thermal energy source has an energy density of 20-60 $\mu W/cm^2$. Researchers have also attempted to harvest energy from humans. As per [16], it includes two types of energy sources: active and passive. Examples of active sources include finger motion and walking, which are capable of generating 2.1 $mW$ and 5W, respectively [16]. In terms of passive sources, breathing is a typical example. Specifically, breathing generates 0.83W; however, at 50% conversion efficiency, only 0.42W is harvested. Hence, human is only used as an energy source to charge small devices.

Recently, Wireless Power Transfer (WPT) has become a promising alternative to overcome the energy constraint of sensor nodes. Interestingly, it allows sensor nodes to be batteryless, and thus they have a much smaller form factor. WPT has three realizations [48]: inductive coupling, RF radiation and magnetic resonant coupling. Inductive coupling generates energy via magnetic field. Specifically, an alternating current generates a magnetic field at a transmitter which in turn induces a voltage at an energy receiver. This technology has a high energy efficiency but a low energy transmission range. For example, reference [49] reports an energy efficiency
of approximately 70% and an energy transmission range of only three centimeters. Magnetic resonant coupling uses two resonant coils. These coils are tuned such that both the sender and receiver have the same resonant frequency. This technique has a high efficiency and was first demonstrated by Kurs et al. [50]. In their experiments, they showed a 60W bulb light located two meters away being charged with an efficiency of 40%. Kurs et al. [51] further developed this technique to charge multiple devices at the same time. In their experiment, a source charges two devices simultaneously. They found that increasing the number of devices improves charging efficiency. For example, when the charging distance is 200 centimeters and there are two devices, the efficiency is about 68%. In contrast, when there is a single device, the efficiency is only about 52%. RF radiation transfers energy from a transmitting antenna to a receiving antenna via RF waves. As per [48], it includes two types: directive RF power beamforming and non-directive RF power transfer. Directive RF power beamforming usually use laser. It has a long energy transmission range and high energy efficiency. However, one main limitation is that it requires line-of-sight (LoS) between a source and a destination [49]. On the other hand, non-directive RF power transfer does not require LoS between an energy transmitter and receiver. It includes energy from analog/digital TV, AM/FM radio, WiFi signal and cellular network [43]. However, it has low energy density; i.e., $0.2 \text{nW/cm}^2 - 1\mu\text{W/cm}^2$ [43]. Further, its energy efficiency is also low. As per [52] the energy efficiency is only 1.5% when the receiver is 30 centimeter away from the transmitter. Nevertheless, as shown in [53], the authors design a prototype sensor platform that harvests energy from digital-TV broadcast.

WPT applications can be categorized into near or far field [48]. Near-field applications usually employ inductive coupling and magnetic coupling. In [54], the authors employ inductive coupling to remotely charge medical implants in humans. In [55], the authors design an online electric vehicle system to charge vehicles via inductive coupling. Their results show an output power of 100 kW with an efficiency of 80% over a 26 centimeters power transmission range. The authors of [23]
1.2 Problem Space

Data gathering is a critical operation in WSNs. It impacts sensing quality and node lifetime. Specifically, the sensing rate of a sensor node determines the sensing quality. One example is a monitoring application for the military [65] that requires high sensing rates to ensure real-time data delivery. Thus, in order to achieve the highest sensing quality, sensor nodes need to increase their sensing rate. However, higher sensing rates result in higher energy consumption, which in turn reduces node lifetime. For example, assume that a sensor node has a lifetime of one year when its sensing rate is 1 pkt/s. If the rate is increased to 2 pkt/s, its energy consumption rate doubles. Consequently, its lifetime reduces. Fair data gathering rate is important to many monitoring applications. One typical example is a forest fire detection system [66]. In particular, a forest fire detection system requires comprehensive information of every sampled or monitored point in order to detect fire threats. If one sensor

use Unmanned Aerial Vehicles (UAV) to wirelessly charge sensor nodes via magnetic resonant coupling. Researchers have also considered using magnetic coupling to charge medical implants [56][57]. Far-field charging applications use RF or laser. An example is the charger from Powercast, which operates with a frequency of 915 MHz, has a power sensitivity of -11.5 dBm and conversion efficiency of 55% [58]. Another example that uses RF to implement wireless charging is the Wireless identification and sensing platform (WISP). It harvests energy from a 900 MHz radio frequency identification (RFID) reader [59]. WISP has been used to develop sensor nodes with a camera [60]. In terms of laser, the authors of [61] use a laser beam to charge sensors. In particular, the authors transfer a laser beam to a distributed light field that covers sensor nodes with a solar cell. Hence, multiple nodes can harvest energy from light. Its energy transmission efficiency reaches 98% over 50 meters. In [62], the authors charge a smartphone equipped with solar cells via a light beam. Table 1.5 compares these WPT techniques.
Table 1.5: Comparison of WPT technologies.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Energy efficiency (Distance)</th>
<th>Advantage</th>
<th>Disadvantage</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inductive coupling</td>
<td>80% (26cm) [55]</td>
<td>Low power loss, simple implementation</td>
<td>Very low energy transmission range, charging one device each time.</td>
<td>Knee implants [54], electric vehicles [55]</td>
</tr>
<tr>
<td>Magnetic resonant coupling</td>
<td>40% (2m)</td>
<td>No requirement for LOS, no interference with neighbors, one-to-many charging</td>
<td>Low energy transmission range, requiring the same operating frequency between a transmitter and a receiver.</td>
<td>UAV [23], medical implants [56][57]</td>
</tr>
<tr>
<td>Non-directive RF power transfer</td>
<td>1.5% (30cm)</td>
<td>No location information requirement, one-to-many charging</td>
<td>Low energy efficiency</td>
<td>WiFi charging system [63], Powercast [58], WISP [64]</td>
</tr>
<tr>
<td>Laser</td>
<td>70% (50m) [49]</td>
<td>Low power lost during transmission</td>
<td>Require LoS, high cost, harmful to human</td>
<td>Laser beam charging [61], AutoCharge [62]</td>
</tr>
</tbody>
</table>
node has a low data gathering rate, a remote center may not take immediate action when a fire occurs.

A key motivation of this thesis is that existing works on data gathering do not consider energy harvesting and WPT together. In particular, as solar is uncontrolled and unpredictable, there will be some sensor nodes with a low energy harvesting rate or a high flow rate. Consequently, they may have insufficient energy to maintain a given Quality of Service (QoS). These nodes are called bottleneck nodes.

Henceforth, this thesis considers using WPT technology to upgrade the energy of bottleneck nodes. There are three key problems: (i) Auxiliary Chargers (AC) placement to maximize the flow rate at the sink (ACP-MF), (ii) ACs placement to maximize the minimum sensing rate (ACP-MM), and (iii) RF energy sharing among nodes to maximize the minimum sensing rate (RFES-MM). These problems are presented in the subsections to follow.

1.2.1 Auxiliary chargers placement to maximize flow rate at the sink (ACP-MF)

The objective is to extract the maximum amount of data from a WSN. A key novelty is the use of ACs to upgrade a subset of sensor nodes to boost their energy supply and thus increase their data forwarding rate. The problem at hand is to determine the placement of a finite number of static Auxiliary Chargers (ACs) with WPT ability. Note, an AC could be an Acroname Garcia robot [67] equipped with a Powercast energy transmitter [58]. As the total cost of an AC can exceed USD $2,000, there are significantly smaller number of ACs as compared to the number of sensor nodes; e.g., a MicaZ mote only costs USD $99.00 [4].

Figure 1.3 illustrates the ACs placement problem. Node $A$ and $C$ are sources that generate 1 and 6 pkt/s, respectively. Nodes $B$, $D$ and $E$ have sufficient energy to forward 3, 2 and 10 pkt/s, respectively. There are two sinks: Sink 1 and Sink 2. Upon inspection, without ACs, the total flow at the sinks is 3 pkt/s. Now consider
the case with two ACs. The problem is to determine suitable locations to deploy these ACs. The aim is to improve the recharging rate of some nodes such that the sinks observe a higher flow rate. In this example, the search space contains $\binom{5}{2} = 10$ possible node pairs. Assume the chosen nodes are $A$ and $D$. Source $A$ increases its data generation rate to 13 pkt/s. Node $D$ has sufficient energy to forward all data from source $C$. Consequently, the total flow rate of the sinks increases to 19 pkt/s.

![Figure 1.3: An example WSN with two ACs and two sinks.](image)

The main factors to consider in this problem include (i) the set of nodes to be upgraded by ACs, (ii) routing, which governs link load and hence the energy consumption of nodes, and (iii) interference, which governs the set of links that can transmit simultaneously and hence their capacity.

### 1.2.2 Auxiliary Chargers placement to maximize the minimum sensing rate (ACP-MM)

This problem is similar to ACP-MF but incorporates source rate fairness. In particular, although sinks may receive the maximum flow, some sources may not transfer data to the sink or have low transmission rates. Consequently, the sensing area in which these sources are located will have less than ideal monitoring rate. Let us consider Figure 1.4. Node $A$ and $B$ are sources. The label next to nodes indicates the number of packets they can sense, transmit and/or receive. Notice that each
node has a different energy harvesting rate. By inspection, the sensing rate of node $A$ and $B$ are 2 and 4 pkt/s, respectively. This is because source $A$ is limited by node $C$, which can only transmit and receive 2 pkt/s. Similarly, node $B$ is bounded by the energy available at node $D$. In this example, the sources have an unequal transmission rate. Consequently, the sink will experience a lower sensing quality from the region monitored by source $A$.

Figure 1.4: An example WSN to show the relationship between energy consumption and maximum data rate.

Thus, in order to maximize the minimum sensing rate, a key challenge is to determine the set of nodes to be upgraded with an AC. In the foregone example, one AC is deployed next to node $C$, and thereby, increasing its relaying rate to 4 pkt/s, which results in node $A$ improving its sensing rate to 4 pkt/s. Consequently, the WSN has better sensing quality after the AC is deployed; i.e., more data is received by the sink, which allows it to form a better picture of the environment.

1.2.3 RF energy sharing to maximize the minimum sensing rate (RFES-MM)

This problem is inspired by the fact that RF signal carries both information and energy. In particular, the problem jointly considers energy sharing between nodes and data routing in a WSN comprising of nodes that harvest energy from both solar and RF. Each sensor node can be a RF energy transmitter and a receiver. Further, nodes
use a time-switching architecture \cite{68} or “harvest-and-transmit” approach, where a proportion of a unit time is dedicated to energy transfer/reception and data transmission. The aim is to maximize the min flow rate of sources. The main decision is to determine the tradeoff between the time used for energy harvesting/transfer and data transmission.

To illustrate the problem at hand, consider Figure 1.5. Assume source \(A\), \(B\) and node \(C\) harvest 10, 10 and 8\(\mu\)J/s from solar, respectively. Their energy consumption rate is 1\(\mu\)J/bit. Assume that a link scheduler assigns link A-C, B-C and C-Sink to their own slot and activates the links for 0.2, 0.2 and 0.4 seconds, respectively. Assume the link capacity is 30 bits/s, meaning these links have a data rate of 6, 6 and 12 bits/s, respectively. However, as the energy of node \(C\) is 8\(\mu\)J, it receives and transfers at most 8 bits/s. Consequently, given these conditions, without energy sharing, the max-min rate is only 4 bits/s. However, node \(A\) and \(B\) have not exhausted their energy. Indeed, the max-min rate is limited by the energy of node \(C\). Now consider the case where nodes share energy. Assume the transmit power is 10\(\mu\)W and there is no energy loss during energy transmission. Assume that source \(A\) and \(B\) each transfers 2\(\mu\)J/s to node \(C\). Specifically, both source \(A\) and \(B\) spend 0.2 second transferring energy to node \(C\). The remaining 0.8 second is used for data transmission. Consequently, the max-min rate increases to 6 bits/s because node \(C\) receives a total of 4\(\mu\)J from both node \(A\) and \(B\).

![Figure 1.5: A WSN example with energy and data transfer denoted by dotted and solid arrows, respectively.](image)

The key challenges are to determine (i) the time portion used by nodes to transfer and receive energy to/from their neighbors, which dictate the amount of available
1.3. Contributions

This thesis aims to solve ACP-MF, ACP-MM and RFES-MM. The main contributions are as follows.

1.3.1 ACP-MF

ACP-MF is the first problem that considers using WPT capable ACs to augment the energy supply of sensor nodes with poor energy harvesting or high energy consumption rate in order to improve the max flow rate at one or more sinks. It is modeled as a Mixed Integer Linear Program (MILP) with the objective to maximize the total flow rate at multiple sinks. ACP-MF unfortunately is a NP-hard problem; see Section 3.1. Thus, the formulated MILP can only be used to solve small problem instances. To this end, three novel heuristic algorithms are proposed; i.e., Path, Tabu and LagOP. Briefly, the Path algorithm aims to recharge all nodes on the shortest path. Based on Path, the Tabu algorithm employs Tabu search to deploy ACs. The LagOP algorithm uses Lagrangian relaxation [69] and sub-gradient optimization to upgrade a set of sensor nodes. Experiment results show that Path, Tabu and LagOP are able to respectively attain 97.00%, 99.40% and 91.65% of the max flow calculated using MILP. In large networks, Tabu has the best performance among the three proposed algorithms. The max flow of LagOP and Path reaches 95.59% and 95.78% of the rate obtained by Tabu, respectively.

1.3.2 ACP-MM

ACP-MM extends ACP-MF but has a different objective: maximize the minimum sensing rate of sources. ACP-MM is also modeled as a MILP. Further, in order
to solve large scale WSNs instances, two greedy algorithms are proposed. The first, called greedy node deployment (GND), iteratively upgrades a node that yields the highest increase in max-min rate. However, a key problem is that it has to search all non-upgraded nodes, and thus incurs a high computational cost. To this end, a second approach, called one unit energy deployment (OUED) algorithm, is proposed. In particular, it uses a relaxed version of the MILP to share one unit energy among all non-upgraded nodes in each iteration. OUED then upgrades the node with the most share. Compared to GND, OUED has lower run time complexity. Experimental results show that GND and OUED are able to respectively attain 99.34% and 98.23% of the max-min rate calculated by MILP in small networks. In addition, as a comparison, the Path algorithm in Section 3.3.1 is employed to calculate the max-min rate. As a result, the max-min rate obtained by Path only archives 46.07% of the max-min rate calculated by MILP. In large networks with 300 nodes, GND results in higher max-min rate than OUED. The average gap between GND and OUED is only 0.035 kb/s. The max-min rate of Path becomes zeros. The running time of GND and OUED in large network are also considered. On average, the running time of OUED only attains 1.744% that of GND.

1.3.3 RFES-MM

Lastly, RFES-MM is the first work that jointly considers energy sharing, sensing rate fairness, data routing and link scheduling together. The problem is formulated as a Linear Program (LP); its objective is to maximize the minimum sensing rate subject to energy, flow, link capacity and active time constraints. A heuristic algorithm called CHMM is proposed to reduce its running time in large problem instances. In Section 5.4, experiments show that, on average, the max-min rate increases by 13.98%; i.e., from 44.93 to 51.21 kb/s, when each sensor node has RF energy transfer ability. In addition, the average gap between the max-min rate obtained by LP and CHMM is 2.56 kb/s. Lastly, Section 5.4 also studies the following parameters: num-
ber of nodes, number of sources, transmit power, conversion efficiency and number of sinks. As we will see later, these parameters influence the time used by nodes for energy transfer/reception; equivalently, they impact data transmission time.

1.4 Publications

The work in this thesis has resulted in the following papers:


1.5 Thesis Structure

1. *Chapter 2*. This chapter surveys works on wireless charging and data gathering. In particular, the works on wireless charging include mobile chargers dispatch strategies and static chargers deployment. The works on mobile chargers contain single and multiple mobile chargers case. The works on static chargers deployment contains single and multi-hop energy transfer scenarios. In addi-
tion, the works on data gathering include throughput maximization and fair rate allocation.

2. Chapter 3. This chapter outlines the ACP-MF problem and shows it is an NP-hard problem. It then outlines the Path, Tabu and LagOP algorithms. In addition, this chapter studies the impact on the max flow when the following parameters are varied: increasing number of nodes, number of sources, number of ACs and node degree.

3. Chapter 4. This chapter proposes the ACP-MM problem and outlines two algorithms: GND and OUED. Next, it analyses several properties of GND and OUED. Lastly, this chapter discusses the influence of some key parameters, including the number of nodes, number of sources, number of ACs and node degree.

4. Chapter 5. This chapter outlines the RFES-MM problem and CHMM algorithm. It then explores the impact of the number of nodes, transmit power, number of sources, conversion efficiency and number of sinks.

5. Chapter 6. This chapter concludes the thesis, and provides a summary of research outcomes and future research directions.
Chapter 2

Literature Review

This chapter reviews past works on wireless charging and data gathering; its structure is elucidated by Figure 2.1. Briefly, based on the type of chargers, the works on wireless charging are divided according to their use of mobile or static chargers. The latter also includes single and multi-hop energy transfer scenarios. Works on data gathering are categorized into two parts according to their objective: throughput maximization or rate fairness allocation. A summary of relevant works is presented in Section 2.3.

2.1 Wireless Charging

To date, numerous past works have employed wireless charging technologies to provide energy to sensors. A typical motivation of these works is to construct a WSN that operates perpetually or to extend its lifetime. Here, lifetime is defined as when the first node depletes its energy. The following subsections review works that use mobile and static chargers.
2.1. Wireless Charging

Figure 2.1: Chapter 2’s structure
2.1. Mobile Chargers

Many works have considered rechargeable wireless sensor networks (rWSNs) comprising of four key parts: one or more mobile chargers (MCs), a service station, one sink and numerous sensor nodes. A typical MC consists of a mobile robot and a wireless energy transmitter. One example is an Acroname Garcia robot with a Powercast wireless energy transmitter [70]. The service station provides energy to MCs. As per [48], there are two types of rWSN model. One model is shown in Figure 2.2. A MC travels and refills the battery of nodes before returning to the service station to recharge itself. The service station commands the MC; in particular, it controls its path and sojourn time at each location. Sensor nodes monitor and report their energy state to the service station. Sensor nodes also monitor their surroundings and transfer data to the sink for further processing. The other model is shown in Figure 2.3. Compared to the first model, this model employs a MC to act as both a wireless charger and data collector. The authors of [48] call this MC a hybrid mobile charger (HMC). Specifically, the HMC stops to charge one or more sensors. Sensors transfer their sensed data to the HMC via one or multi-hops. After a charging tour, the HMC returns to the service station to deliver data and recharge itself. Section 2.1.1.1 and 2.1.1.2 review works that use single and multiple chargers, respectively.

2.1.1.1 Single Charger

Peng et al. [70] design a Greedy algorithm that selects $k$ nodes with the shortest lifetime and sets the second shortest node lifetime as the target network lifetime $T$. They consider the charging sequence of a MC, which includes its charging path and charging time at each stop location. The MC’s charging path is found using the Traveling Salesman Problem (TSP) solver proposed in [71]. The proposed Greedy algorithm ensures the MC has sufficient energy to return to the service station.
If a charging sequence is found, the target network lifetime is achievable and the algorithm extends the target lifetime to the node with the next highest lifetime. However, the Greedy algorithm charges a node as long as possible and does not consider whether the network lifetime of other nodes can be extended to $T$. Therefore, in order to avoid energy wastage, the authors propose Greedy Plus, an algorithm that uses binary search along with the Greedy algorithm to provide a more exact
2.1. Wireless Charging

Li et al. [72] jointly consider data routing and energy replenishment. Their objective is to maximize network lifetime. Specifically, a service station also acts as the sink. Sensor nodes periodically send data to the sink via the Collection Tree Protocol (CTP) [73]. They also inform the sink their residual energy level and consumption rate. Expected Transmission Count (ETX) [74] is used to estimate the energy consumption rate due to retransmissions. Next, the authors consider two types of routing: energy minimum and energy balanced. The former is concerned with the data transmission path that consumes the least energy. However, this means nodes on the path consume energy faster. In contrast, energy balanced routing forwards packets across as many nodes as possible to amortize energy expenditure. However, this increases overall energy consumption. Thus, a charging-aware routing is employed to consider both energy-minimum and energy-balanced routing. Specifically, the authors select the nodes to be charged via their future energy consumption rates. In order to reduce the energy consumption of the whole network, energy-minimum paths are used as frequently as possible. The authors then employ binary search to determine the maximum target network lifetime and allocate energy for each node. Next, the authors consider charging sequence. Specifically, nodes are sorted by their lifetime in ascending order. However, if the MC travels to charge nodes that only require a small amount of energy, the energy incurred by traveling is wasted. Therefore, the authors avoid charging these nodes. The charging time of a node with the largest lifetime is iteratively merged into the charging time of the node with the smallest lifetime until the battery of the minimum lifetime node is fully charged. As a result, there are fewer nodes to be charged in a tour. Lastly, the authors model the problem as a LP that aims to maximize network lifetime. Its constraints include energy and flow conservation. Further, the returned solution ensures the total charging time of the MC is smaller than the network lifetime.

Zhao et al [75] jointly consider energy replenishment and data gathering. In particular, they employ a HMC that selects several sensor nodes as anchor points
to charge these nodes. Further, the HMC gathers data from nodes near anchor points via multi-hop communications. As there is a trade-off between number of anchor points and data gathering latency, the authors propose to find the maximum number of anchor points and a tour length that does not exceed a given threshold. The charging tour is determined by a TSP solver. In order to select nodes to be charged, all nodes are sorted by their residual energy level in increasing order. The authors then use binary search to determine the maximum anchor points such that the total length of the resulting charging tour does not exceed a given threshold.

Next, the authors consider data gathering performance. Specifically, they aim to maximize the amount of data gathered from each sensor. The problem is cast as a utility maximization problem. The key constraints include flow conservation, energy and link capacity. Note that they also consider wireless interference among nodes. They then devise a distributed algorithm that employs Lagrangian multiplier and dual decomposition, see [76] for a key reference, to find the optimal data rate for each node and the flow rate over each link.

Shi et al. [77] propose to maximize the vacation time over the renewable cycle time. In particular, a renewable cycle includes three parts: traveling, charging and vacation time. They correspond to the traveling time of the MC, time used to charge sensor nodes, and how long the MC spends at the service station. The authors then prove that the optimal charging path of the MC is the shortest Hamiltonian cycle. The constraints include flow conservation, energy conservation and residual energy threshold at sensor nodes. As the problem is non-linear, the authors reformulate the problem via the change-of-variable technique and approximate the non-linear constraints linearly. The resulting problem is then solved using CPLEX [78]. A limitation of this work is that it only charges one node each time. Thus, when the node density is high, sensor nodes may not be charged in a timely manner.

Xie et al. [79] build on [77] and consider high node density. Specifically, compared to the MC in [77], the one in [79] charges multiple nodes simultaneously. As shown in Figure 2.4, the network area is partitioned into hexagonal cells. Each one
is sized according to the charging range of a MC. Further, the MC stops at the center of each cell represented by the black point. The authors use a TSP solver to compute a traveling path for the MC. The authors prove that the problem is a nonlinear problem (NLP) and propose a provably near optimal solution. In particular, the authors transform the NLP problem to a 0-1 Mixed Integer NLP (MINLP). Reformulation and Linearization (RLT) [80] is employed to convert the MINLP into a MILP which is solved by CPLEX [78]. However, one shortcoming is as follows. Cell $A$ covers three nodes in Figure 2.4. If cell $A$ is moved to the left, it is able to cover four nodes and the MC does not have to travel to the center of cell $B$. Therefore, if the MC knows the locations of nodes before a charging tour and the minimum number of cells are used to cover all nodes, the number of stop locations of the MC may be decreased to reduce the total traveling path.

![Figure 2.4: Hexagonal cells covering a network.](image-url)
Li et al. [81] also consider jointly optimizing energy replenishment and data gathering. The WSN in Figure 2.5 contains nodes that are deployed regularly in a square topology. A HMC is employed to charge nodes and collect data. With regard to the charging path of the HMC, the selected nodes form a continuous square wave shape. An example is shown in Figure 2.5. The initial path of the HMC is represented by a black line. The HMC then goes through each anchor node and stays for a period of time to recharge its surrounding nodes. Next, in order to charge nodes more evenly, the authors shift the path of the HMC. As shown in Figure 2.5, the dotted line represents the next path of the HMC. After multiple consecutive charging tours, the charging path goes through all nodes; i.e., all nodes are charged by the HMC over time. After determining the charging path, the authors assume that each sensor transfers its data to its nearest nodes on the charging path. They then propose to assign a transmission rate to each sensor that maximizes the network utility. A deterministic solution can be found because routing is fixed in each charging tour.

Figure 2.5: An example charging tour.
2.1. Wireless Charging

He et al. [82] propose an on-demand charging algorithm that considers both temporal and spatial properties. Specifically, they show that the simplest scheduling method is First-Come-First-Serve (FCFS). However, this method only focuses on the arrival sequence of the charging requests from nodes. It ignores spatial condition such as the position of sensor nodes, and may cause a MC to move back and forth between sensor nodes. Thus, the authors propose Nearest-Job-Next-with-Preemption (NJNP) to determine the sequence of nodes to be charged. In particular, each node sends a request when its residual energy is lower than a threshold. As a departure from past works, their MC does not process requests at the end of a charging tour. Instead, the MC processes a request after charging a node and re-selects the next node to charge. In order to address the back and forth movement problem when using FCFS, the MC stores and sorts requests from nodes in a service pool by their order of arrival. Subsequently, the MC selects nodes that are spatially closest to the requesting node.

Fu et al. [1] consider a WSN with a mobile RFID reader that is used to charge numerous WISPs. They aim to find the stopping positions, and corresponding duration, of the RFID reader with the aim to minimize the total charging time. They then show that this problem can be modeled as a LP. However, as a RFID reader can be deployed anywhere, solving the LP results in a high computational overhead. Thus, the authors devise a method to reduce the number of stopping locations. In particular, they use Welzl’s work [83] to construct the Smallest Enclosing Space (SES) to cover all WISPs, as shown in Figure 2.6. Next, the authors discretize the charging power. Specifically, the black points in Figure 2.6 represent WISPs. The authors then use concentric circles, shown in dotted circles in the figure, to represent different charging rate levels due to different distances. Locations that have a similar charging rate become a candidate stop location. In order to further reduce the number of stop locations, the authors use Lloyd’s algorithm [84] which is a $k$-means clustering algorithm to merge the stop locations of the RFID reader. Lastly, the authors can use a LP solver to solve the problem.
Xie et al. [85] aim to maximize the ratio of vacation time over the renewable cycle time. However, compared to [77], this work employs a HMC. Further, the HMC can stop anywhere in the WSN. In order to reduce candidate stop locations, the authors narrow the roaming area of the HMC into a SED. Further, similar to [1], the authors construct concentric circles to represent different wireless charging rates. Then, the authors consider data gathering. Concentric circles are also constructed to represent the energy consumption rate when data is transferred. As a result, two kinds of concentric circles partitions are present. They give rise to sub-areas that are used as candidate stop locations. They then use the shortest Hamiltonian cycle to connect all stopping points in order to construct a charging path. Next, the authors employ change-of-variable to reformulate the NLP to a LP with five constraints. First, any feasible solution must ensure that the distance between a node and the HMC is within recharging range. The second constraint is the standard flow conservation. Third, the total energy consumption is bounded by the sum of transmission and reception cost. Fourth, the remaining energy at the end of each charging cycle must be larger than the minimum energy required by a node to
2.1. Wireless Charging

![Diagram](image)

Figure 2.7: (a) A charging path computed by the method in [2] that has the minimum length, (b) A naive charging tour where the MC stops at each node

operate. Fifth, the charging energy must be larger than the energy consumption in each charging cycle.

Li et al. [2] consider balancing the energy usage due to traveling and charging. In particular, the authors assume that the energy consumed to charge each node is constant. One MC with finite energy capacity is used to charge nodes. They then propose to find a charging tour that covers the maximum number of nodes. In order to solve the problem, the authors use Particle Swarm Optimization (PSO) [86]. Specifically, as shown in Figure 2.7(a), the authors construct a circle centered at each sensor node. The circle’s radius is the charging range threshold of the MC. The authors employ Lin-Kernighan (LKH) [87] as the TSP subroutine to obtain the stopping points in the search area and construct a charging tour. When the tour is obtained, the authors check the energy level of the MC. If it is not sufficient to charge nodes, the authors remove stop points which results in reducing a smallest number of covered nodes. However, as shown in Figure 2.7(b), this method requires a MC to charge nodes at the boundary of circles. Therefore, this method has a lower charging efficiency than a method that visits each node to charge; see Figure 2.7(b).

Table 2.1 compares these works from six aspects: objective, key decision vari-
ables, key constraints, charger type, problem model and control method. Specifically, in terms of objective, only references [70] and [72] aim to maximize network lifetime. The remaining works consider data gathering [75][81], data routing [77][79][85] or charging efficiency [82][1][2]. Further, with regards to charging sequence, references [70][72][75][77][79][81] and [82] determine a charging path by selecting a set of sensor nodes. However, in references [1][85] and [2], the stop location of MCs can be arbitrary. In addition, only references [70] and [2] consider an MC’s energy constraint. In terms of charger type, works such as [79][1] and [85] employ a charger that is able to simultaneously charge multiple sensors. References [75][81] and [85] employ a HMC to charge nodes. Lastly, there are only two works that study distributed solutions; i.e., [75] and [82].

Nevertheless, past works that use a single charger have the following drawbacks. Specifically, references [70][82][1] and [2] only devise a charging strategy of a MC. They ignore data routing and data gathering. References [72][77] and [79][85] jointly consider data routing and wireless charging. However, they neglect data gathering rate allocation among sensor nodes. Only references [75] and [81] consider data gathering rate. However, reference [81] ignores wireless interference among sensor nodes. Lastly, all works using a single charger to charge nodes have a common drawback where the single charger with a finite energy capacity cannot provide timely charging service in a large scale WSN. This thus motivates researchers to use multiple chargers; the next topic.
Table 2.1: A summary of past works on using single MC

<table>
<thead>
<tr>
<th>Work</th>
<th>Objective</th>
<th>Key Decision Variable</th>
<th>Key Constraint</th>
<th>Charger Type</th>
<th>Problem Model</th>
<th>Control Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peng et al. [70]</td>
<td>Maximize network lifetime</td>
<td>Charging sequence</td>
<td>MC’s energy capacity</td>
<td>MC, one-to-one</td>
<td>N/A</td>
<td>Centralized</td>
</tr>
<tr>
<td>Li et al. [72]</td>
<td>Maximize network lifetime</td>
<td>Charging sequence, data routing</td>
<td>Total charging time threshold</td>
<td>MC, one-to-one</td>
<td>LP</td>
<td>Centralized</td>
</tr>
<tr>
<td>Zhao et al. [75]</td>
<td>Maximize network utility</td>
<td>Charging sequence, data routing, data rate</td>
<td>Link capacity based on wireless interference</td>
<td>HMC, one-to-one</td>
<td>NUM</td>
<td>Distributed</td>
</tr>
<tr>
<td>Shi et al. [77]</td>
<td>Maximize the ratio of vacation time to renewable cycle time</td>
<td>Charging sequence, vacation time, data routing</td>
<td>Residual energy threshold at sensors</td>
<td>MC, one-to-one</td>
<td>NLP</td>
<td>Centralized</td>
</tr>
<tr>
<td>Xie et al. [79]</td>
<td>Maximize the ratio of vacation time to renewable cycle time</td>
<td>Charging sequence, vacation time, data routing</td>
<td>Residual energy threshold at sensors</td>
<td>MC, one-to-many</td>
<td>NLP</td>
<td>Centralized</td>
</tr>
<tr>
<td>Li et al. [81]</td>
<td>Maximize network utility</td>
<td>Data rate</td>
<td>Energy, flow conservation</td>
<td>MC, one-to-one</td>
<td>NUM</td>
<td>Centralized</td>
</tr>
<tr>
<td>He et al. [82]</td>
<td>Reduce charging delay</td>
<td>Charging request sequence</td>
<td>Location of the MC</td>
<td>MC, one-to-one</td>
<td>-</td>
<td>Distributed</td>
</tr>
<tr>
<td>Fu et al. [1]</td>
<td>Minimize total charging time</td>
<td>Charging sequence</td>
<td>Working energy threshold at sensors</td>
<td>MC, one-to-many</td>
<td>LP</td>
<td>Centralized</td>
</tr>
<tr>
<td>Xie et al. [85]</td>
<td>Maximize the ratio of vacation time to renewable cycle time</td>
<td>Charging sequence, vacation time, data routing</td>
<td>Link capacity and wireless interference</td>
<td>HMC, one-to-many</td>
<td>NLP</td>
<td>Centralized</td>
</tr>
<tr>
<td>Li et al. [2]</td>
<td>Cover the maximum number of nodes</td>
<td>Charging location</td>
<td>MC’s energy capacity</td>
<td>MC, one-to-one</td>
<td>LP</td>
<td>Centralized</td>
</tr>
</tbody>
</table>
2.1. Wireless Charging

2.1.1.2 Multiple Chargers

Zhang et al. [88] consider a 1-D WSN and numerous MCs with energy capacity constraint. Sensor nodes with the same energy capacity are uniformly distributed on a line topology. A base station acts as an energy source for MCs and data sink of sensor nodes. The authors define payload energy and overhead energy as the amount of energy used by MCs to charge sensors and move, respectively. Assume that each node has the same energy consumption rate. They then propose to maximize the ratio of payload to overhead energy. One key constraint is that each node must not exhaust its energy. The authors devise a charging strategy for multiple MCs. Specifically, all MCs depart the service station at the same time. The line topology is divided into multiple segments, each charged by a MC. The intersection of two adjacent segments is called a rendezvous point. Assume that MC A charges nodes in a given segment and stops at a designed rendezvous point a. The other MCs are charged to their full battery capacity via MC A at rendezvous point a. The MCs except A continue to move to the next rendezvous point. MC A waits for other MCs to return to rendezvous point a. When the MCs return to point a, MC A evenly distributes its energy to MCs so that they have sufficient energy to travel to the next rendezvous point. Next, the authors consider nodes that have non-uniform energy consumption rate. In particular, to save traveling cost, nodes are clustered into groups via their charging cycle. For example, the authors assume two nodes are charged every four and two time slots, respectively. Each charging tour only charges two nodes in order to reduce traveling cost.

Wang et al. [89] propose two problems: minimize the number and total traveling cost of MCs in order to achieve perpetual operation. Specifically, the authors construct a network in a hierarchical manner. Next, based on the concept of Named Data Networking (NDN), each MC sends packets to update its routing table. Sensor nodes then transfer their energy status to the MC. They then consider minimizing the number of MCs. In particular, they employ Bernoulli process to model the
2.1. Wireless Charging

energy consumption of a node; i.e., the probability that a node consumes one unit energy in one unit time slot is $p$. In order to obtain perpetual operation in $n$ time slots, a sensor node must ensure its replenished energy is larger than its consumed energy. The authors calculate the probability of attaining perpetual operation and then obtain the minimum number of MCs when the probability is larger than 0.99. Next, the authors address the problem of minimizing the total traveling cost of multiple MCs in order to achieve perpetual network operation. Specifically, the authors assume that MCs start and end at the service station when they complete a charging cycle. Each charger has a time constraint on its charging tour. Each node is visited only once. They then model the problem as a Multiple Traveling Salesmen Problem with Deadlines (MTSPD) [91]. The authors propose a heuristic algorithm to calculate the charging sequence of nodes. Two parameters are considered: traveling time to the next node and remaining lifetime of the current node. A weighted sum of the two parameters is used to determine the priority of a node in a charging sequence. The node with the smallest weighted sum has the highest charging priority. Lastly, HMCs communicate with each other to determine which nodes are not charged and thus avoid repeatedly charging the same node.

Madhja et al. [92] propose a Distributed Coordination Local Knowledge (DCLK) protocol that uses multiple MCs to charge nodes. Specifically, nodes are uniformly deployed in a circular area where at the center is the sink. The MCs are evenly distributed and nodes are charged fully. They divide the charging process into two phases: coordination and charging. In the coordination phase, the authors evenly split the network area into sectors. Each MC is responsible for charging a sector. According to the current energy level of a MC and the MC’s energy consumption rate, the authors determine whether to enlarge or reduce the sector for each MC. In the charging phase, the authors divide each sector into small sub-sectors with the same width. In order to determine the charging sequence of sub-sectors, the authors assign a weight to each sub sector. The weight is a product of the number of nodes whose energy is under a threshold and the amount of energy to be charged. The
sub sector with the maximum weight is charged first.

Dai et al. [93] propose to minimize the number of MCs in order to achieve perpetual operation. In particular, they assume that each node has the same energy consumption rate. Each MC completes a renewable cycle to charge sensor nodes. Four constraints are proposed: (i) the vacation time of MC’s threshold, (ii) energy conservation, (iii) perpetual operation of nodes, and (iv) MC’s energy capacity. In order to reduce the complexity of the problem, the authors develop an approximation algorithm that considers energy capacity constraint, but omit constraint (ii) and (iii). Then, Dai et al. employ the solution to the Distance Constrained Vehicle Routing Problem (DVRP) [94] to calculate the traveling path of each MC.

Erol-Kantarci et al. [95] consider a timely and efficient charging strategy in a smart grid monitoring system. Each MC starts at a service station and then charges multiple sensor nodes at a point called a landmark. The authors aim to select the minimum number of landmarks. They then model the problem as an Integer Linear Program (ILP) with the following constraints. First, a MC at a landmark has to serve as many nodes as possible. Second, the energy capacity of a MC must be larger than the energy requirement of the nodes it charges. When the location of landmarks is determined, the authors begin to group landmarks for each MC. In order to reduce the total traveling path of MCs, the authors group landmarks via their proximity to the service station of each MC. In order to reduce the traveling cost of each MC, the shortest Hamiltonian cycle is employed.

Table 2.2 compares works that consider multiple MCs. It includes five aspects: objective, key decision variables, key constraints, control method and charger type. Specifically, all works except [92] focus on improving charging efficiency. Compared to [88][89] and [93], works [92] and [95] only optimize the charging path of MCs. In terms of constraint, references [88][92] and [93] employ energy-constrained MCs. By contrast, references [89] and [95] assume MCs have infinite energy. Lastly, with regard to charger type, only the authors of [95] consider charging multiple sensors simultaneously.
2.1. Wireless Charging

These reviewed works have the following drawbacks. First, reference [88] considers charging nodes only in a line topology. Second, in [93] and [95], each sensor node has the same energy consumption rate. Reference [88] assumes each node has a fixed energy consumption rate. In addition, reference [89] assumes the energy consumption rate of a node is related to its location. However, in practice, nodes near the sink usually forward more data than those farther away from the sink, meaning the nodes near the sink have a higher energy consumption rate.
Table 2.2: A summary of past works that use multiple MCs.

<table>
<thead>
<tr>
<th>Work</th>
<th>Objective</th>
<th>Key Decision Variables</th>
<th>Key Constraint</th>
<th>Control Method</th>
<th>Charger Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang et al. [88]</td>
<td>Maximize the ratio of payload energy to overhead energy</td>
<td>Charging sequence of each MC</td>
<td>MC’s energy capacity</td>
<td>Centralized</td>
<td>MCs, one-to-one</td>
</tr>
<tr>
<td>Wang et al. [89]</td>
<td>Minimize total traveling cost of MCs</td>
<td>Charging sequence of each MC, number of MCs</td>
<td>A WSN with perpetual operation</td>
<td>Centralized</td>
<td>MCs, one-to-one</td>
</tr>
<tr>
<td>Madhja et al. [92]</td>
<td>Devise a protocol to prolong network lifetime</td>
<td>Charging path of each MC</td>
<td>MC’s energy capacity</td>
<td>Distributed</td>
<td>MCs, one-to-one</td>
</tr>
<tr>
<td>Dai et al. [93]</td>
<td>Minimizing the number of MCs</td>
<td>Charging sequence of each MC, number of chargers</td>
<td>MC’s energy capacity, perpetual operation WSN</td>
<td>Centralized</td>
<td>MCs, one-to-one</td>
</tr>
<tr>
<td>Erol-kantarci et al. [95]</td>
<td>Minimizing the number of landmarks</td>
<td>Charging location of MCs</td>
<td>Energy conservation, one node covered by one MC</td>
<td>Centralized</td>
<td>MCs, one-to-many</td>
</tr>
</tbody>
</table>
2.1.2 Static Chargers

As per [52][48], there are three key scenarios in works that assume static chargers; namely, point, path and multi-hop provisioning scenarios. In point provisioning, both nodes and chargers are static. Chargers upgrade the energy of a node over one hop. By contrast, path provisioning considers deploying static chargers to charge mobile nodes over one-hop. In multi-hop provisioning, static chargers are able to transfer its energy over multiple hops. Further, nodes are able to share energy with each other. Sections 2.1.2.1 and 2.1.2.2 review works that use single and multi-hop energy transfer, respectively.

2.1.2.1 One Hop Energy Transfer

He et al. [52] propose to deploy the minimum number of RFID readers to ensure WISP tags have perpetual operation in point and path provisioning scenarios, respectively. In point provisioning, to cover all tags, the authors partition a network into equilateral triangles and a reader is placed on each vertex; see Figure 2.8. The point with the minimum charging rate is shown to be the center of each triangle. Using this fact, the authors calculate the minimum number of readers used to supply tags with energy. In the path provisioning case, the authors assume tags are distributed uniformly. They assume tags move according to the random way point mobility model [97]. Next, they investigate the problem of deploying the minimum number of readers. A key constraint is that the accumulative recharging rate of a tag in the whole sensing area is larger than its power consumption. This rate is determined by the recharging rate from readers that a tag encounters. To solve the problem, they employ equilateral triangles to cover the whole sensing area and place readers on the vertices of triangles. They then calculate the maximum side length of the triangles such that the number of readers is minimum subject to tags receiving sufficient recharging rate. Note, as tags move to power-rich areas to harvest energy and then go through power hungry areas, the path provisioning case uses
fewer readers as compared to the point provisioning case.

Chiu et al. [98] aim to select the position of chargers in order to maximize the survival rate of nodes on a grid map. The authors assume nodes move with certain regularity. An infra-red counter is deployed at each intersection to record the number of passing nodes, and is used as the weight of each intersection. Specifically, the authors predict the position where nodes deplete their energy. The intersections where the nodes deplete their energy are sorted according to the weight of intersections. Finally, the first $N$ highest weight intersections are selected as chargers deployment points.

Talla et al. [63] demonstrate the use of access points (APs) to charge battery-free sensor nodes. They first showed that data transmissions are insufficient to power sensor nodes. This is because during silent periods the capacitor of sensor nodes leaks. In order to address this problem, APs improve channel occupancy by injecting so called power packets, i.e., UDP broadcast packets. Moreover, to ensure capacity remains high, APs with power packets are scheduled to transmit together.

Li et al. [99] propose to deploy the minimum number of RF chargers to meet the energy requirement of battery-free sensor nodes. Unlike past works, this work
2.1. Wireless Charging

considers a new recharging model. In particular, a node can rely on the energy from multiple chargers. A key consideration is that these chargers have a different distance and their signals may combine or cancel each other. They then model the problem as a NLP. In order to solve the problem, they devise a Greedy algorithm that finds a grid position that enables the most number of nodes that meet their minimum energy requirement. As the Greedy algorithm results in local optima, they devise a PSO approach. A key challenge is that for a big surveillance area, it becomes impractical to solve. So the authors use the method in [100] to group sensor nodes that are close together into a cluster. Each cluster is then solved independently using PSO.

Doost et al. [101] use a fixed RF energy transmitter (ET) to charge nodes and define the charging time of a node as the time taken to reach a given voltage level. They then use a node’s charging time as a routing metric. A source selects the route with the lowest charging time to forward data to the sink. In each time slot, all nodes on the selected route have a common charging time. After charging, the remaining time is used for data transmission. Their aim is to maximize the throughput of a selected route.

Nintanavongsa et al. [102] developed a wireless energy transfer protocol for sensor nodes that can harvest energy from two frequencies. Sensors of type I harvest energy from the 614 MHz digital TV (DTV) band while those in type II use the 915 MHz ISM band. All sensors communicate with an ET via a 915 MHz control channel. ETs operate on both the 614 and 915 MHz band. Two frequencies are used because the authors consider DTV signal to be intermittent and is thus insufficient to be used solely to power sensor nodes. Consequently, sensors of type I have to receive energy from an ET. To do this, an ET broadcasts a Request to Charge (RTC) packet when the channel is free. All sensors that hear the RTC respond with a Clear to Charge (CTC) packet to the ET. Next, the ET uses the received signal strength of each sensor’s CTC packet to estimate its received power. In particular, the authors obtain the charging and discharging curve. Specifically, the charging
curve is a function of charging time and received signal strength. On the other hand, the discharging curve indicates the relationship between time and current voltage at node. The authors then propose to maximize the total transferred energy at the ET by determining the charging time of the ET. A key consideration is that the operating voltage of sensors reaches a given threshold. Another consideration is that the sum of charging rate for the two types of sensors subtracted by their discharging rate is maximized.

In [103], Naderi et al. design a CSMA/CA MAC protocol for a RF energy harvesting WSN with several ETs and sensors. Specifically, when the energy level of a node drops to a given threshold, it broadcasts an energy request packet. ETs that receive this packet send back a pulse. A key consideration is that multiple ETs are likely to be located at different locations and thus their signal may combine or cancel each other. Thus, based on the received signal strength of the transmitted pulse, the node estimates the distance to ETs and groups ETs via the phase of their arriving signal. Next, the node optimizes the frequency to assign to each group that maximizes the energy transfer from ETs in each group. The key constraints are that the spectrum shape of the two groups does not overlap and the bandwidth used is within that of a node’s energy harvesting circuit. The authors then determine the upper charging threshold and charging duration. In addition, the authors consider energy request packets to have a higher access priority than data packets. The data packets from a node with ample energy have a higher priority to access channel than the data packets from a node with low residual energy.

Ju et al. [104] consider a wireless powered communication network where there is a hybrid access point (H-AP) that functions as a power source and data collector. They propose a “harvest-then-transmit” protocol whereby nodes harvest RF energy from the H-AP in the downlink and then transmit their data to the H-AP in the uplink. They propose to maximize the common throughput or max-min rate of nodes by determining the optimal time to receive energy and transfer data for each node. The problem is solved via convex optimization [105].
2.1. Wireless Charging

In [106], the authors consider a full-duplex H-AP that simultaneously receives information from nodes and broadcasts energy to nodes in a time block that includes multiple slots. They assume that node $i$ sends information to the H-AP at time slot $i$ and receives energy from the H-AP in the remaining slots. They propose to maximize the weighted sum-rate of all nodes by determining the transmission power of the H-AP at each time slot and duration of each time slot. There are three constraints. First, the sum of all time slots cannot be larger than a time block. Second, the energy broadcast by the H-AP in a time block has a given upper bound. Third, the H-AP’s transmission power is bounded by a peak value. The authors then show the problem is non-convex. To transfer the problem into a convex optimization problem, the authors introduce a new variable that represents the energy or power broadcasted by the H-AP in a time slot. Further, the authors show that the objective function is a monotonically increasing function of the new variable under a given duration of time slot. The authors then solve the problem using the standard Lagrangian duality framework.

Zhao et al. [107] study the “harvest-then-transmit” protocol in a communication network that includes a power source, a user node and an information receiver. The key idea is how to balance the time used to harvest energy and transfer data in order to maximize the throughput at information receiver. In particular, in each time slot $T$, let $t$ be the time used by the user node to harvest energy from the power source. The user node uses the remaining time $T - t$ to transmit data. The authors find that transmission power monotonically increases with $t$. On the other hand, the information error rate monotonically decreases with $t$. Further, as there is an upper bound of the transmission power and information error rate, the authors compute the upper and lower bound of $t$ and then numerically search the optimal value of $t$ to maximize the throughput.

From Table 2.3, past works using single hop energy transfer focus on improving charging efficiency; see [52][98][99][102] and [103]. Some consider optimizing data gathering; see [63][101][104][106] and [107]. They usually consider chargers deploy-
2.1. Wireless Charging

ment, e.g., [52][98][99], or develop a time allocation strategy to control energy or data for specific objectives; e.g., [102][103][104][106] and [107]. Note, references [104][106] and [107] consider the trade-off between energy reception and data transmission time at nodes. All works except [99] and [103] assume that the received energy from multiple chargers can be additive. Note, the common drawback of single hop energy transfer is its energy transmission range. For example, in these works, chargers are static once they are deployed. Thus, sensors that are far away from chargers have low energy harvesting rates.
Table 2.3: A summary of past works on single hop energy transfer.

<table>
<thead>
<tr>
<th>Work</th>
<th>Objective</th>
<th>Key Decision Variable</th>
<th>Key Constraint</th>
<th>Control Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>He et al. [52]</td>
<td>Minimize number of chargers</td>
<td>Charger location</td>
<td>Perpetual operation WSNs</td>
<td>Centralized</td>
</tr>
<tr>
<td>Chiu et al. [98]</td>
<td>Maximize the survival rate of nodes</td>
<td>Charger location</td>
<td>Number of chargers</td>
<td>Centralized</td>
</tr>
<tr>
<td>Talla et al. [63]</td>
<td>Demonstrate WiFi charging</td>
<td>-</td>
<td>Power threshold to boot up energy harvesting function</td>
<td>Centralized</td>
</tr>
<tr>
<td>Li et al. [99]</td>
<td>Minimize number of chargers</td>
<td>Charger location</td>
<td>Minimum energy requirement of sensors</td>
<td>Centralized</td>
</tr>
<tr>
<td>Doost et al. [101]</td>
<td>Maximize throughput of the selected route</td>
<td>Charging time at nodes</td>
<td>Node operation voltage threshold</td>
<td>Centralized</td>
</tr>
<tr>
<td>Nintanavongsa et al. [102]</td>
<td>Maximize the total transferred energy at the ET</td>
<td>Charging time of the ET</td>
<td>Highest duty cycle, lower charging threshold</td>
<td>Centralized</td>
</tr>
<tr>
<td>Naderi et al. [103]</td>
<td>Design CSMA/CA based MAC protocol</td>
<td>Charging time at nodes, operating frequency of ETs</td>
<td>Different spectrum shape among groups of ETs</td>
<td>Distributed</td>
</tr>
<tr>
<td>Ju et al. [104]</td>
<td>Maximize the min rate of nodes</td>
<td>Time duration of receiving energy and transferring data</td>
<td>One time block</td>
<td>Centralized</td>
</tr>
<tr>
<td>Ju et al. [106]</td>
<td>Maximize throughput</td>
<td>Transmission power at H-AP, time duration of each time slot</td>
<td>One time block</td>
<td>Centralized</td>
</tr>
<tr>
<td>Zhao et al. [107]</td>
<td>Maximize throughput</td>
<td>Time duration of receiving energy and transferring data at the node</td>
<td>One time block, information error rate</td>
<td>Centralized</td>
</tr>
</tbody>
</table>
2.1.2 Multiple Hops Energy Transfer

Kaushik et al. [108] experimentally show that compared to direct or one-hop energy transfer, two-hop energy transfer via a relay node results in a higher received energy at a given destination. Specifically, the energy source is a HAMEN RF synthesizer HM8135 with a transmit power of +13 dBm at 915MHz. The relay and end nodes are constructed by a Powercast P1110 energy harvester and a Mica2 mote with a +6.1 dBi antenna. The transmit power of the relay node is +3 dBm. Note, the relay node has two modes: ON and OFF. In the ON mode, the end node receives energy from the source and relay node. In the OFF mode, the end node only receives energy from a source node; i.e., Kaushik et al. consider one-hop energy transfer. The authors change the relay from ON to OFF mode to compare the voltage level under the same charging time. Their experiment results show that two-hop energy transfer reaches a higher voltage level during the same charging time than one-hop energy transfer.

Watfa et al. [109] compare three multi-hop energy transfer techniques: store and forward, direct flow and hybrid. The first technique, i.e., store and forward, means a node receives energy and stores energy in its rechargeable battery. It then transfers the energy to the next hop when its battery is fully recharged. One shortcoming is energy loss during storage. The second technique, namely direct flow, allows a node to directly transmit received energy to the next node. Consequently, there is no energy storage loss. However, the energy loss during transmission accumulates. The last technique, i.e., hybrid technique, combines the previous two techniques. Nodes employ direct flow to transfer energy via $M$ hops and then stores the energy at the $M$-th node. This process is repeated until the energy is transmitted over a given number of hops. As per their simulation, the value of $M$ is related to the total number of transmission hops. Further, they show that the hybrid technique has a higher number of transmission hops than the other two techniques.

Krikidis et al. [110] consider a three-node network that includes a source, relay
and destination node. The relay node is a half-duplex node that receives data/energy from a source node and then transfers data/energy to a destination node. Thus, it has two models: harvesting and relay. In the harvesting model, the relay harvests RF energy from the source. On the other hand, in the relay model, the relay node transmits data of the source. The authors study a greedy switching policy whereby the relay node transmits data to the destination when it has sufficient energy. They show via a finite Markov chain the policy is close to the optimal switching policy where the relay node knows the channel coefficients and energy state of its battery. Lastly, they propose to maximize the number of data transmission at the relay node and model the problem as a MILP subject to energy constraint at the relay node.

Fouladgar et al. [111] consider reusing some of the received energy from incoming links for transmissions. Specifically, a node’s available energy is the sum of energy received from incoming links, after properly accounting for path loss, and energy obtained from solar. For a given link \((i, j)\), a transmission rate of \(r_{ij}\) will require \(p_{ij}\) Joules. Thus, a key energy conservation constraint is that the total energy used on all outgoing links, each with a different data rate, must exceed the available energy at a node. A key decision variable is the transmission power of each link, which is dictated by the total flows traversing said link. In particular, a high transmit power means more harvested energy and hence allowing a neighbour to transmit at a higher rate. The objective is to maximize the source rate in unicast and multicast scenarios subject to the said energy conservation constraint and also the standard flow conservation constraint. The problem is solved via the standard tools of convex optimization.

Rault et al. [112] aim to deploy the minimum number of static chargers to ensure nodes operate perpetually. The location of chargers is restricted to the location of nodes. Compare to [99], the authors assume that each node can transfer RF energy to its neighbors. In order to calculate energy loss over multi-hops, the authors associate an energy loss coefficient to each link, which is proportional to the length of the link. For any pair of nodes, a source selects the route with the lowest energy.
loss to transfer energy to a receiver. The authors employ a modified Dijkstra's algorithm to construct a tree that incurs the minimal energy transfer loss. Each charger acts as a root of a charging tree. Next, the authors model the problem of deploying the minimum number of chargers as a MILP that satisfies the following three constraints: (i) a node only has one charger, (ii) each charger has limited battery capacity, and (iii) each charging tree is disjoint.

Chin et al. [113] propose to minimize the superframe length by controlling data routing and active time of links to meet the flow demand among wireless routers and the energy demand of sensors. In particular, they consider a two-tiered RF energy harvesting network comprising of wireless routers and sensors. There are two link types: data links between routers and energy links between a router and a sensor. The data links are used to transfer data to meet flow demand between a pair of routers. The authors assume that a sensor harvests RF energy from data link. However, if the active time of data links is not sufficient, routers transmit power packets; similar to [63]. The authors then model the problem as a LP to minimize the superframe length. The aim is to (i) solve for the active time of links, which determines their capacity and also the amount of energy supplied to sensors, and (ii) also generate the route of source-destination pairs, which determines the traffic load on each link. The LP has four key constraints. The first is flow conservation between routers. The second is that total traffic flow on each link does not exceed the link’s capacity; recall that a link’s capacity is determined by its active time. Next, for each sensor node, its harvested energy from both data and energy links must be larger than its energy demand. Lastly, the superframe length is at most one.

Xiang et al. [114] consider a multi-hop network powered by an AP. In particular, the AP is responsible for injecting or distributing power to nodes. Moreover, the injected power must meet the demand of nodes. The formulated problem aims to construct an energy distribution tree as well as determine the routing of data flows. They assume nodes use magnetic resonance coupling; this allows nodes to
2.1. Wireless Charging

distribute power to multiple nodes with an efficiency that increases with the number of receivers. They first propose to minimize the total power injected by the AP that meets each node’s power requirement. However, the efficiency ratio, i.e., the fraction of transmission power harvested by a node, is dependent on the number of receivers, making the problem non-convex. They then construct an LP as follows. Let a virtual node $v$ represents the transmission of power from a node $i$ to all combinations of its neighbors. That is, if node $i$ has two neighbors, then node $i$ belongs to two virtual nodes corresponding to it charging one neighbor or both neighbors simultaneously. Each node is connected to the virtual node in which it is the transmitter, and there is an edge connecting each node to virtual nodes in which it is a receiver. The objective remains the same. The decision variable is the amount of power transferred to each virtual node such that for a given node $i$, its received minus transmitted power must exceed its requirement. However, the number of virtual nodes grows according to the node degree and hence, the resulting LP is only solvable for small instances. To solve the problem, they first derive the dual of the LP and highlight that the virtual nodes in the optimal solution have a negative cost. They then present an algorithm that preferentially selects the virtual node with the highest price and adds as many receivers as possible subject to its total cost remaining negative. The algorithm stops when all virtual nodes have a negative cost. Lastly, they present a joint data and energy routing formulation. The goal is to minimize the power emitted by the sink subject to three constraints. First, the demand of each source-destination pair is satisfied. Second, each link is not activated for more than one time unit. Third, the traffic flow on each link is within capacity.

Gurakan et al. [115] also consider energy transfer or cooperation with the aim of reducing the total transmission delays of all links. Each link $l$ has delay $\frac{t_l}{c_l - t_l}$, where $t_l$ is the amount of traffic on link $l$ and $c_l$ is its capacity. The capacity of a link is determined by its transmission power. Each node has two energy sources: solar and RF energy from its neighbors. The authors consider three scenarios. First, they consider the one-slot case and assume $t_l$ for each link $l$ is given. The problem is to
determine the transmission power $p_l$ and also the amount of energy to transfer to each neighbor in a given slot. The problem is convex, and via Karush-Kuhn-Tucker (KKT) conditions, the authors show that more power is allocated to links with high traffic or noise, and if there is energy cooperation, nodes with low traffic load should transfer their energy to those with a higher traffic load. They then present an algorithm that first solves the corresponding Lagrangian function for the optimal dual variables for the case without energy transfer. The dual variable for each energy link is then revised iteratively for the energy transfer case. The algorithm activates each energy link in turn, and the optimal dual variable for each energy link that satisfies energy causality is computed; specifically, the authors derive the optimal energy causality condition at the optimal point. The problem then is to adapt the dual variable to meet this condition. If no such dual variable is found, the algorithm transfers any previously transferred energy back to the transmitting node. The algorithm continues until all links meet the said condition. The authors proved that their algorithm converges to the optimum value. The second scenario concerns the same problem but over $T$ slots. Moreover, a node is able to transfer any excess energy in time slot $k$ to $k + 1$. The authors show that the multiple slots case can be transformed into the single slot case. Specifically, they create $T$ replicas of the network graph. Then an energy link with perfect transfer efficiency is added between the replicas of the same node. The resulting network is then equivalent to the single slot case and can be solved using the algorithm for the one-slot case. The last scenario is to jointly consider $p_l$, amount of energy transferred to each neighbor and also the amount of traffic routed on a given path; hence, $t_l$ is now a decision variable where its value is dependent on the total traffic being routed over link $l$. It is worth noting that each source has a fixed demand. The only requirement is that the total traffic transmitted by a source must arrive at the corresponding destination. They propose an iterative algorithm that assigns more power to links that cause the biggest increase in the objective function. Also, it iteratively moves traffic from the path that causes the minimum increase to the objective function to one that
causes the highest increase in the objective function. Lastly, the algorithm ensures
the optimal amount of energy is transferred.

Table 2.4 summarizes works that use multiple hops energy transfer. References [108][109][110] purely consider multi-hop energy transmission. References [111][114][115] jointly consider data routing and power allocation on links. They consider a receiver simultaneously obtaining information and RF energy from its received signal. However, reference [116] shows that current circuits cannot directly obtain RF energy from decoded information.

2.2 Data Gathering

To date, there are a lot of works that consider data gathering. These works usually optimize data flow to achieve some specific objectives. In regards to the focus of this thesis, only works that aim to maximize throughput or fairness are relevant to the problems in Chapter 1.

2.2.1 Throughput Maximization

Typical approaches include the following: (i) Data routing. An example is [117] where the authors consider un-splittable flows in order to maximize the flow at a sink. They then show that the problem can be solved via a standard Max-Flow Min-Cut algorithm; e.g., [118]. The maximum flow, however, is restricted by the energy of nodes, and (ii) Node placement. Researchers have considered placing nodes to improve connectivity, network lifetime and throughput; e.g., [119][120] and [121]. Note, Section 2.1.2 has already reviewed works on deploying chargers; see [52][98][99][112]. Thus, this subsection focuses on works that deploy relays and sinks.

Flushing et al. [121] consider a WSN on a 2-D grid. Each grid point represents a candidate location for relay nodes. The objective is to maximize throughput and reduce end-to-end delays by selecting locations to place relay nodes. The authors
Table 2.4: A summary of past works on multiple hops energy transfer.

<table>
<thead>
<tr>
<th>Work</th>
<th>Objective</th>
<th>Key Decision Variable(s)</th>
<th>Key Constraint(s)</th>
<th>Network Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kaushik et al. [108]</td>
<td>Experimentally show multi-hop energy transfer</td>
<td>ON-OFF mode switched at the relay</td>
<td>-</td>
<td>Three nodes</td>
</tr>
<tr>
<td>Watfa et al. [109]</td>
<td>Design a hybrid technique to improve charging efficiency</td>
<td>Three techniques</td>
<td>-</td>
<td>Line</td>
</tr>
<tr>
<td>Krikidis et al. [110]</td>
<td>Maximize a relay node’s data transmission times.</td>
<td>Harvest-relay model switching</td>
<td>Energy constraint at the relay node</td>
<td>Three nodes</td>
</tr>
<tr>
<td>Fouladgar et al. [111]</td>
<td>Maximize the source rate in unicast and multicast scenarios</td>
<td>Transmission power of each link</td>
<td>Energy and flow conservation</td>
<td>Unicast and multicast network with one source</td>
</tr>
<tr>
<td>Rault et al. [112]</td>
<td>Minimize the number of static chargers</td>
<td>Location of chargers</td>
<td>Perpetual operation, charger energy capacity</td>
<td>Static nodes with chargers</td>
</tr>
<tr>
<td>Chin et al. [113]</td>
<td>Minimize superframe length</td>
<td>Active time of each link</td>
<td>Flow conservation, link capacity, energy require-</td>
<td>Wireless routers and RF energy harves-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ting nodes</td>
<td>tng nodes</td>
</tr>
<tr>
<td>Xiang et al. [114]</td>
<td>Minimize the total power injected by an AP</td>
<td>Transmission power of AP, data routing</td>
<td>Interference among nodes, link capacity</td>
<td>One AP and multiple nodes</td>
</tr>
<tr>
<td>Gurakan et al. [115]</td>
<td>Minimize the total transmission delays of all links</td>
<td>Data routing, power on each link, data rate</td>
<td>Energy, flow conservation, link capacity</td>
<td>An energy harvesting network</td>
</tr>
</tbody>
</table>
then formulate it as a Mixed Integer Program (MIP); the objective is to minimize link load, the number of relay nodes and the degree of each relay node. The decision variables are the data flow on each link and the position of each relay node. The constraints include flow conservation, link capacity and number of relay nodes. However, the MIP has high computational complexity because of the large number of candidate locations. Thus, the authors propose two heuristic algorithms to reduce the number of candidate locations. The first algorithm only deploys relay nodes on the straight line between two nodes. The second heuristic algorithm uses the density of sensor nodes. In particular, the authors uniformly divide the network into several sub-regions. Let $\delta$ represent the distance between two grid points on a 2-D grid. There are $k$ given possible values of $\delta$. Note, these values are sorted by ascending order. They compute the difference between the sensor node density at a sub-region and the average sensor node density over the whole sensing area. The authors then obtain the absolute value of the difference. Lastly, the authors select a higher $\delta$ to a sub-region with a high absolute value. Otherwise, the authors allocate a lower $\delta$ value to a sub-region with a low absolute value.

In [122], Flushing et al. extended their work in [121] to consider uncertain data generation rates. A discrete set is used to record all possible data generation rates. Sensor nodes select a value from this set as their data generation rate. The authors then employ a robust optimization approach. Specifically, the authors propose a regret metric to evaluate the robustness of selected relay nodes locations. The regret metric has three parts: the minimum flow cost of the selected relay node locations, the minimum flow cost without relay nodes and the minimum flow cost given the optimal relay node locations. If the regret metric is zero, the selected relay node locations are optimal for all data rates in the set. Thus, they transfer the problem to minimizing the maximum value of regret. They evaluate all possible placements of relay nodes and all possible data generation rates to obtain the maximum relative regret. In order to reduce computational complexity, the authors propose an approximation algorithm to reduce the number of data generation rate scenarios.
2.2. Data Gathering

Specifically, based on relay node placements, the authors calculate *favoured nodes*, which correspond to sensor nodes with routing paths to any sinks that pass through at least one relay node. Also, the data generation rate of *favoured nodes* is set to the highest value. In contrast, for non *favoured nodes*, their data generation rate is set to the lowest value. They then calculate the *regret* of each selected relay node location to obtain the maximum *regret* value. The authors evaluate all possible placements of relay nodes to obtain the minimum maximum regret. However, this incurs a high computational complexity. Consequently, the authors propose a genetic algorithm (GA) approach. The selected relay node locations act as individuals in GA. Next, the authors design genetic operators. The crossover operator is a randomly selection of relay node locations from both parents. The mutation operator is that a current relay node location is replaced by another candidate location within communication range of the relay node. The method for calculating *regret* is similar to their previously proposed approximation algorithm in [123].

Ali et al. [124] consider uniformly distributed sensor nodes that are divided into several groups. Each group has a cluster header (CH) that is responsible for collecting data from sensors. The CH then transfers collected data to an AP. The authors aim to find the optimal location of a given number of CHs such that the throughput at the AP is maximum. In order to maximize throughput, the authors aim to minimize the total packet transmission time from sensors to the AP by deploying CHs. The network is comprised of a number of sensor nodes and an AP. The sensors are uniformly distributed in a circular area with an AP deployed at the centre. The authors propose a genetic algorithm to minimize the total packet transmission time from sensors to the AP. An individual is a possible deployment of all CHs. The fitness function calculates the total packet transmission time. The crossover operator randomly selects the location of CHs from parents to construct children. The mutation operator randomly changes CH locations. The authors assume a specific maximum number of generations. Each generation has 20 individuals. The fitter an individual is the more chance it will be selected as a parent. Two parents are
selected. Next, the two parents use the crossover and mutation operator to generate children (next generation). This process repeats until the specified maximum number of generations is reached. The fittest individual in the final generation is the solution.

Deng et al. [125] aim to maximize the total collected data by deploying sinks in an online fashion. That means sinks are deployed one by one. Sensor nodes have no information about the number, position and data capacity of a sink until the sink is deployed. Each sensor only knows the distance between itself to its nearest sink. The authors propose a near-optimal online algorithm via a primal-dual approach. First, the authors design an offline dual problem. The objective is to maximize the total data that sensor nodes transmit. The first constraint is that sensor nodes cannot exhaust their energy. The second constraint is that a sink only can receive data that is smaller than its data capacity. The decision variables are the data flow rate from a sensor to a sink and sensor node’s transmission power. Next, the authors transfer this offline dual problem to an online primal problem via a primal-dual approach. In order to maximize the competitive ratio, the authors aim to maximize the minimum distance and minimize the maximum distance between sinks and nodes. Specifically, the authors reduce the sink location region to a SES that covers all sensor nodes. In order to minimize the maximum distance between sinks and sensor nodes, the authors use a NLP to represent the problem of minimizing the maximum distance. In order to change the NLP to LP, the authors discretize the SES using the algorithms in [1] such that there is a limited number of zones for sinks in the SES. When a sink is deployed in a zone, the distance between the sink and a node has a lower and upper bound. Thus, the NLP is reduced to a LP with a decision variable corresponding to the zone to be selected by each sink.

Sridharan et al. [126] consider a data gathering tree where leaf nodes are data generators and the root is the sink. Each node has a limited bandwidth capacity and interference with its neighbors. The authors propose two problems. The first is to maximize the minimum data gathering rate subject to capacity and flow conservation
constraints. Each child computes its available bandwidth and transfers it to their parent. The parent considers the available bandwidth of itself and its children and obtains the minimum one to transfer it to its parent. The minimum available bandwidth of the root is the minimum of all available bandwidth. After that, the authors propose to maximize the sum rate of all sources. They then use a Lagrange dual sub-gradient method. In addition, they propose a heuristic that determines bottleneck nodes and then sets the rate of sources to that of bottleneck nodes.

Table 2.5 compares works on throughput maximization. Note, all works, except [124], jointly consider data routing and throughput maximization. References [125][126] also consider sensing rate. However, throughput maximization has a drawback. In particular, it results in unfair rate allocation among sources. For example, in order to maximize the throughput at the sink, the source that is far away from the sink has a low sensing rate. This is because these nodes consume more energy to transfer data to the sink than nodes near the sink. Consequently, the sensing quality of the area covered by the source is low. The sink cannot immediately receive data from the area. Therefore, a key problem is fair rate allocation among sources.

2.2.2 Fair Rate Allocation

As mentioned, fair rate allocation is required to ensure high sensing quality; i.e., data is collected “fairly” from all sensed areas. Past works can be categorized into two types: (i) trade-off between network lifetime maximization and max-min rate allocation, and (ii) lexicographic max-min (LMM) rate allocation under a given network lifetime. In particular, in order to maximize network lifetime, sensor nodes with high energy are preferred to transfer data. However, it results in unfair rate allocation; i.e., nodes with low energy transfer less data to ensure network lifetime. With regards to LMM rate allocation, a standard approach is to employ LP iteratively; see [127] for more information.

Zhu et al. [13] study the trade-off between maximizing network lifetime and
### 2.2. Data Gathering

Table 2.5: A summary of works on throughput maximization

<table>
<thead>
<tr>
<th>Work</th>
<th>Objective</th>
<th>Key Decision Variable(s)</th>
<th>Key Constraint(s)</th>
<th>Online?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flushing et al. [121]</td>
<td>Maximize throughput and reduce end-to-end delays</td>
<td>Data routing, location of relay nodes</td>
<td>Number of relays, flow conversation, link capacity</td>
<td>No</td>
</tr>
<tr>
<td>Flushing et al. [122]</td>
<td>Maximize throughput and reduce end-to-end delays</td>
<td>Data routing, location of relay nodes</td>
<td>Number of relays, flow conservation, link capacity, data generation set.</td>
<td>No</td>
</tr>
<tr>
<td>Ali et al. [124]</td>
<td>Maximize the throughput at the AP</td>
<td>Location of relays</td>
<td>Location of relays is represented by polar coordinates</td>
<td>No</td>
</tr>
<tr>
<td>Deng et al. [125]</td>
<td>Maximize the total data collected by sinks</td>
<td>Location of sinks, data rate, energy consumed by each node</td>
<td>Energy constraint and data capacity at each node</td>
<td>Yes</td>
</tr>
<tr>
<td>Sridharan et al. [126]</td>
<td>Maximize the sum rate of all sources</td>
<td>Data rate at each source</td>
<td>Limited bandwidth capacity with wireless interference, max-min data gathering rate</td>
<td>No</td>
</tr>
</tbody>
</table>
max-min rate allocation as a concave function with a weighted objective. Specifically, the function has two parts. The first part aims to maximize the minimum lifetime; equivalently, this is also equal to minimizing the maximum normalized power dissipated. The second part maximizes the sum of utility functions; each of which represents the utility for a given transmission rate of each sensor. They use the NUM framework to maximize the sum of utility functions, and thereby achieve proportional fairness. A key consideration is that each sensor node has a set of predetermined routes to a sink. Further, each node has a flow conservation constraint. In addition, the total consumed energy at a sensor node cannot exceed its residual energy. The resulting problem is a convex optimization problem. To solve the problem, the authors prove that, for an optimal solution, the nodes with the maximum normalized power dissipated have negative profit. To this end, for a given initial flow rate on each route, the authors calculate the profit of each route and then employ a sub-gradient algorithm to update flow rate to reach the optimal solution.

Lai et al. [14] also consider the trade-off between network lifetime and rate fairness allocation. Specifically, they construct a data gathering tree where leaf nodes generate sensed data and the root is the sink. The authors first propose to maximize network lifetime subject to link capacity and energy of node constraint. To solve the problem, they introduce bit capacity. For a leaf node, based on its residual energy, its bit capacity is the maximum data that it transfers. For a relay node, its bit capacity is the minimum value between its maximum data that it transfers and the total received data of its children. The authors then calculate lifetime of a node via its bit capacity and link capacity. After that, they formulate a convex optimization problem where the goal is to maximize the product rate of all sources. They then devise a low-complexity algorithm to compute the optimal rate allocation. The algorithm iteratively reduces the number of constraints. In each iteration, each source updates its bit capacity by the bit capacity of their parents. Note, nodes having the same parent evenly share bit capacity of the parent. The algorithm then deletes the parent node. Consequently, all sources directly connect
2.2. Data Gathering

to the sink. For each source, the allocated data rate is calculated by its current bit capacity and link capacity.

Srinivas et al. [15] consider a two tiered network comprising of Mobile Backbone Nodes (MBNs) and regular nodes. They aim to maximize the minimum throughput of regular nodes by placing a fixed number of MBNs and assigning regular nodes to each MBN. The authors assume that regular nodes select one MBN to transfer information and those that belong to different MBNs do not interfere with each other. The throughput of a regular node is related to its distance to a MBN and the number of regular nodes associated to the MBN. Initially, in order to reduce the infinite candidate locations of MBNs to finite candidate locations, the authors define a 1-center location, i.e., the location of a MBN that minimizes the farthest distance from any assigned regular nodes in its area. All 1-center locations become candidate locations of MBNs. Then, the authors select the location for MBNs such that the farthest distance from any regular node to the nearest MBN is minimized. Next, the authors consider the problem of assigning regular nodes to each MBN and then cast the problem as an Integer Max-Flow problem. In particular, the authors construct a flow graph where a source connects all regular nodes and all MBNs connect to a destination. The capacity of links between the source and a regular node is one. The capacity of the link between a MBN and the destination is related to two factors: the number of nodes assigned to it and its data transmission range. The authors then find all assignments by connecting regular nodes and MBNs in order to maximize the minimum rate among all nodes. In addition, the authors propose two low complexity heuristic algorithms. The first algorithm aims to reduce the number of candidate locations of MBNs. The authors only consider node location and a pair of regular nodes as the candidate locations of MBNs. The second algorithm places the first MBN on any regular node and all regular nodes transfer data to the first MBN. Next, the second MBN is deployed on the regular node that is farthest away from the first MBN. All regular nodes select the nearest MBN to transfer data. This process repeats until all MBNs are deployed.
Numerous works advocate LMM rate allocation and throughput maximization under a given network lifetime; see [127] for more information. Briefly, an LMM rate allocation is defined as follows. Let $\tilde{g} = \{\tilde{g}_1, \tilde{g}_2, \cdots, \tilde{g}_|V|\}$ be a sorted rate vector. For any other rate vector $\hat{g}$, if there exists a $k$ ($1 \leq k \leq |V|$) as well as $\tilde{g}_i = \hat{g}_i$ $(1 \leq i \leq k - 1)$ and $\tilde{g}_k \geq \hat{g}_k$, then $\tilde{g}$ is an LMM rate allocation.

The authors of [127] propose an algorithm that iteratively computes the value of each LMM rate. Specifically, in each iteration, the authors divide the nodes into two parts. The nodes that have an allocated LMM rate are called determined nodes. Otherwise, they are called undetermined nodes. They then propose two maximize either the common rate (MCR) or single rate (MSR). In MCR, they aim to compute the maximum rate that all undetermined nodes can reach. Note, MCR is equivalent to the LMM rate. The authors model the MCR problem as a LP. In MSR, the aim is to determine the nodes that have the LMM rate. In particular, they consider one undetermined node each time. They use a LP to model the problem of maximizing the rate of an undetermined node. Note, the rate of all remaining undetermined nodes is set to the current MCR. If the maximum rate is equal to the computed MCR, then all nodes have an LMM rate. Finally, all nodes that obtain their LMM rate are set as determined nodes. The following works focus on devising an algorithm to solve either MCR or MSR.

The following works also employ a LP to compute MCR. However, they devise different algorithms to solve the MSR problem. In particular, Hou et al. [128] employ the parametric analysis (PA) technique of [129] to determine the minimum undetermined nodes set in each iteration. For each undetermined node, the authors increase its rate by a small amount. If the common rate of all undetermined nodes decreases, the authors allocate the current LMM rate to the undetermined node. Marasevic et al. [130] predict time-varying recharging rates during $T$ time slots in an energy harvesting WSN. In each iteration, after obtaining the MCR, they then check each node’s energy level at the start of the next slot. If its energy level becomes zero, the node is marked as determined.
2.2. Data Gathering

Liu et al. [131] propose a distributed algorithm that allocates the LMM rate in a solar-powered WSN. All nodes rely on solar power and are organized in a tree topology. Leaf nodes or sources send a control packet containing their maximum data rate and flow identification (ID) to their parent. A parent compares its maximum data rate with the total maximum data rate from all its children. If the parent does not have sufficient data rate, the parent evenly allocates its data rate to each flow from its children. Otherwise, the parent forwards all data from its children. This process is repeated until the control packet arrives at the sink.

Yang et al. [132] propose a distributed protocol to allocate LMM rate to sources in topologies with multiple sinks. Specifically, in each iteration, the authors first determine the MCR among all nodes without a LMM rate via a distributed single-objective optimization problem. Its key constraints include energy, flow conservation and equal rate among undetermined nodes. Next, the authors consider a distributed solution to determine whether the MCR is the LMM rate of a node. In particular, each node locally checks its residual energy. If a sensor node reaches its maximum energy consumption, a sensor node is saturated and multicast a one-hop control packet to all its upstream neighbours. Otherwise, the sensor node is unsaturated. When an unsaturated node receives a control packet from its downstream neighbour, the node changes its state to saturated. Further, if the node receives a control packet from all downstream neighbours, it multicasts a control packet to its upstream neighbors. Lastly, all saturated nodes uses MCR as their LMM rate.

Table 2.6 compares references [13][14] and [15] according to their objective, key decision variables, key constraints and network topology. References [13] and [14] control the data rate of source to achieve their objective. References [15] jointly consider rate allocation and node placement. Table 2.7 compares past works on LMM rate allocation.
2.2. Data Gathering

<table>
<thead>
<tr>
<th>Work</th>
<th>Objective</th>
<th>Key Decision Variable(s)</th>
<th>Key Constraint(s)</th>
<th>Network Topology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhu et al. [13]</td>
<td>Network lifetime maximization and fair rate allocation</td>
<td>Flow rate on each route</td>
<td>Energy and flow conservation</td>
<td>Tree topology</td>
</tr>
<tr>
<td>Lai et al. [14]</td>
<td>Maximize network lifetime, maximize the product rate of all sources</td>
<td>Data rate at each source</td>
<td>Energy constraint, link capacity</td>
<td>Tree topology</td>
</tr>
<tr>
<td>Srinivas et al. [15]</td>
<td>Maximize the minimum throughput of regular nodes</td>
<td>Location and set of MBNs</td>
<td>Assign a node to one MBN, flow conservation</td>
<td>Two-tiered network</td>
</tr>
</tbody>
</table>

Table 2.6: A summary of [13][14] and [15].

<table>
<thead>
<tr>
<th>Work</th>
<th>Solution</th>
<th>Network Topology</th>
<th>Control Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chen et al. [127]</td>
<td>Serial LPs</td>
<td>Sensors and multiple sinks</td>
<td>Centralized</td>
</tr>
<tr>
<td>Hou et al. [128]</td>
<td>LP, PA</td>
<td>A WSN with one sink</td>
<td>Centralized</td>
</tr>
<tr>
<td>Marasevic et al. [130]</td>
<td>Check energy level at the start of a slot to determine node sets</td>
<td>A solar-powered WSN with varying recharging rate</td>
<td>Centralized</td>
</tr>
<tr>
<td>Liu et al. [131]</td>
<td>A node evenly allocates its data rate to each flow when its data rate is not sufficient.</td>
<td>A tree solar-powered WSN</td>
<td>Distributed</td>
</tr>
<tr>
<td>Yang et al. [132]</td>
<td>A node sends a control packet to its upstream neighbors when its LMM rate is determined.</td>
<td>A solar powered WSNs with multiple sinks</td>
<td>Distributed</td>
</tr>
</tbody>
</table>

Table 2.7: A summary of works on LMM rate allocation
2.3 Summary

This chapter has reviewed past works on wireless charging and data gathering. At this point, the reader is reminded of the key problems addressed in this thesis. Recall that the ACP-MF problem considers deployment strategy of a given number of auxiliary chargers in order to maximize the flow rate at one or more sinks. Compared to the ACP-MF problem, the ACP-MM problem aims to deploy ACs to maximize the minimum sensing rate among all sources. Lastly, RFES-MM has the same objective as ACP-MM. However, the problem focuses on determining the energy harvesting time and data transmission time of each sensor node. This thesis differs from these works as follows:

1. Past works on throughput maximization do not consider combining energy harvesting sensor nodes and AC deployment together in order to achieve their objective. For works on wireless charging, references [106][107] and [111] are the only works that aim to maximize throughput. However, references [106] and [107] only consider a simple topology with three nodes. The energy source is a RF energy transmitter. Reference [111] considers a topology with only one source and allocates power to each link. It neglects wireless interference among nodes. Further, no works have considered upgrading resource constrained nodes to improve the maximum flow rate at one or more sinks.

2. Past works on fair rate allocation aim to derive a routing solution. Further, some works consider maximizing the minimum sensing rate in rechargeable WSN; see [130][131][132]. The performance of these works is restricted by sensors with the lowest energy harvesting rate. By contrast, ACP-MM uses ACs to upgrade these nodes to further increase the max-min rate. In terms of works on wireless charging, only reference [104] has the same objective as ACP-MM. However, the network model considered in [104] is different. Specifically, the authors consider one-hop data transmission from source to destination.
3. Compared to the works on multiple hops energy transfer, RFES-MM considers network with arbitrary number of hops. Further, all nodes can be an energy transmitter in RFES-MM. In addition, past works do not consider max-min rate. Compared to works on fair rate allocation, RFES-MM considers multi-hop energy transfer and nodes use a time-switching architecture. In addition, references [104][106] and [107] also consider time allocation for energy and data in one time slot. However, RFES-MM employs multi-hop energy transfer. Further, nodes transfer data via one-hop in [104][106] and [107]. By contrast, in RFES-MM, nodes transfer data over multi-hops. RFES-MM is the first work that jointly considers link scheduling, rate fairness allocation, data routing, multi-hop energy transfer and time switching.
This chapter takes advantage of advances in WPT to address the ACP-MF problem: maximize the amount of data forwarded to a sink where sensor nodes harvest energy from solar as well as from a co-located AC if one exists. This problem is significant for the following reason. After deployment, some sensor nodes may impede the amount of data that arrive at a sink because of their low energy harvesting rate. This has implications on applications that require gathered data to be of high fidelity; i.e., they require as much data as possible from a WSN in order to detect faults [65]. The maximum sensing or flow rate at a sink, however, is determined by the available energy at sensor nodes.

The problem is modeled as a MILP and three novel heuristic algorithms are proposed. Namely, Path, Tabu and LagOp. Briefly, the Path algorithm aims to recharge all nodes on the shortest path. This chapter then presents a Tabu search algorithm. Lastly, based on Lagrangian relaxation [69], this chapter devises an algorithm called LagOP. The rest of the chapter is organized as follows. Section 3.1 introduces key notations and network model. Section 3.2 defines the problem formally. Section 3.3 presents the details of Path, Tabu and LagOP. Section 3.4 reports experimental results. Finally, Section 3.5 concludes the chapter.
3.1 Preliminaries

Consider a rechargeable WSN modeled as a graph $G(V, E)$, where $V$ represents the set of nodes and $E$ represents the set of links between any two nodes in $V$. Let $(i, j) \in E$ represent the link between node $i$ and $j$. Let $\mathcal{S} \subset V$ be the set of sinks; each of which is denoted as $s$. The term $\delta_i$ represents the node degree or number of neighbors of node $i$. Let $\mathcal{S} \subseteq V - \mathcal{S}$ be the set of sources or sensor nodes that generate data. For each node $i$, the set $V_i^- \subset V$ contains all neighbors from which node $i$ receives data. Conversely, the set $V_i^+ \subset V$ denotes neighbors to which node $i$ sends data to.

Let matrix $A$ represent transmission sets; each row corresponds to a link and each column represents a transmission or independent set. Specifically, an entry $a^n_{i,j}$ in $A$ denotes whether a link $(i, j)$ is active in column $n$. For example, if $a^n_{i,j} = 1$, link $(i, j)$ is active in the transmission set $n$. The set of links in column $n$ can be determined using either the protocol or physical interference model [133]. That is, each column denotes the set of links that can transmit simultaneously without interfering with one another as per the said model. The transmission or link schedule is represented by a vector $S = [x_1, x_2, \ldots, x_N]$ where $x_n$ is the active time of column $n$ of matrix $A$ and $\sum_{n=1}^{N} x_n = 1$.

Initially, node $i$ has a battery with energy level $e_i$. Each node has two energy sources: solar and/or WPT. When node $i$ is recharged by a solar panel, its recharging rate is $E_i$ Joule per second. There are $R$# ACs. If an AC is placed next to a node, then its recharging rate increases by $\beta$ Joule per second. Thus, if a node $i$ is recharged by an AC, its total recharging rate is $E_i + \beta$. Without loss of generality, assume the only source of energy consumption is communication. Specifically, a node consumes $\rho$ and $\tau$ Joule for receiving and transferring one bit, respectively. Let $f_{ij}$ and $f_{ji}$ represent the flow rate (bps) over link $(i, j)$ and $(j, i)$, respectively. A binary variable $R_i \in \{0, 1\}$ indicates whether an AC is deployed next to node $i$; i.e., $R_i = 1$ means an AC is deployed next to node $i$. Table 3.1 gives a summary of common notations.
3.2 Problem Formulation

The aim of ACP-MF is to deploy $R^#$ ACs to provide extra energy to $R^#$ nodes in order to maximize the flow rate at the sink. This section models ACP-MF using a MILP that includes five constraints: energy, flow conservation, link capacity, the length of a superframe and number of ACs.

The energy constraint ensures the amount of recharged energy for each node is larger than its consumed energy. Formally, for each node $i \in V - \mathcal{S}$,

$$\rho \sum_{u \in V_i^-} f_{u,i} + \tau \sum_{v \in V_i^+} f_{i,v} \leq R_i \beta + E_i,$$  \hspace{1cm} (3.1)

The expression on the left-hand side represents the total energy consumption of node $i$ whilst the right-hand side gives the total recharged energy of node $i$. Observe that multiple ACs are assigned, up to say $\Phi$, next to a node by revising $R_i$ to take on a value in the range $\{0, 1, \ldots, \Phi\}$. The following constraint for each node $i \in V - \mathcal{S}$
shows standard flow conservation constraint,

$$
\sum_{u \in V_i^-} f_{u,i} = \sum_{v \in V_i^+} f_{i,v}, \quad (3.2)
$$

The expression $\sum_{u \in V_i^-} f_{u,i}$ means the sum of input flow at node $i$. In contrast, the expression $\sum_{v \in V_i^+} f_{i,v}$ represents the sum of output flow at node $i$. Next, the capacity of a link $(i, u)$ is determined by its total active time. Hence, for each link in $E$,

$$
f_{i,u} \leq \sum_{n=1}^{N} a_{i,u}^n x_n C, \quad (3.3)
$$

The maximum time in which all links are active is constrained to one second. Formally,

$$
\sum_{n=1}^{N} x_n = 1, \quad (3.4)
$$

As there are only a finite number of ACs,

$$
\sum_{i \in V - \emptyset} R_i = R^\# \quad (3.5)
$$

The objective function aims to maximize the total flow rate at the sinks,

$$
\text{MAX} \quad \sum_{s \in \emptyset} \sum_{i \in V_s^-} f_{i,s}
\text{ s.t. } (3.1), (3.2), (3.3), (3.4), (3.5)
$$

The decision variables of the MILP include $f_{i,j}$, $R_i$ and $x_n$.

Next, there is a proof stating that ACP-MF is NP-hard. In particular, even for a single source and sink case, the problem is equivalent to solving the NP-hard knapsack problem.

**Proposition 1.** The ACP problem is NP-hard.
3.2. Problem Formulation

Proof. Briefly, in the knapsack problem, there is a set $I$ of $n$ items, i.e., $I = \{1, 2, \ldots, n\}$, in which each item $i \in I$ has a value $b_i$ and cost $c_i$. The knapsack problem, for a given cost constraint $C^\#$, aims to find some subset $S \subseteq I$ that maximizes $\sum_{i \in S} b_i$ subject to $\sum_{i \in S} c_i \leq C^\#$. Consider an instance of ACP-MF as illustrated in Figure 3.1. There is a sink node $t$, a source node $s$ and $|A|$ intermediate nodes, $a_i \in A$, for $i = 1, 2, \ldots, |A|$, between them. Further, there is an edge, with infinite capacity, from $s$ to each $a_i$, and from each $a_i$ to $t$. Note that each $a_i \in A$ represents a node disjoint path in ACP-MF whose nodes can be upgraded to increase the flow from $s$ to $t$. Let the value of $a_i$, denoted as $v(a_i)$, be the increase in the amount of routed data if one or more nodes in the disjoint path $a_i$ are upgraded with an AC. Note, only disjoint paths to ensure each $a_i$ is unique, i.e., it contains nodes different from those in any other $a_j$, for $i \neq j$. Further, the increased flow rate at the sink can simply be computed by taking the sum of flow rate passing through each disjoint path $a_i$, i.e., $\sum v(a_i)$. Let cost $p(a_i)$ represent the total number of upgraded nodes in $a_i$ to achieve value $v(a_i)$. At this point, for a given number of ACs, i.e., $R^\#$, the problem is to find some subset $S' \subseteq A$ that maximizes $\sum_{i \in S'} v(a_i)$, subject to $\sum_{i \in S'} p(a_i) \leq R^\#$. The problem is equivalent to the NP-hard knapsack problem, i.e., $v(a_i)$, $p(a_i)$ and $R^\#$ are equivalent to $b_i$, $c_i$ and $C^\#$, respectively, and thus ACP-MF is also NP-hard.

Before delving into solutions, there are two remarks. First, algorithms can be used to consider the problem of determining the minimum ACs required to meet a given flow rate. Assume a given WSN requires the max flow rate to be $B$ bits/s. Thus, there is the following MILP

$$\begin{align*}
\text{MIN} & \quad R^\# \\
\text{s.t.} & \quad \sum_{s \in \mathcal{E}} \sum_{i \in V_i^-} f_{i,s} \geq B \\
& \quad (3.1), (3.2), (3.3), (3.4), (3.5)
\end{align*}$$
Second, a standard approach that facilitates computational convenience is to introduce two new nodes: a virtual source $\mu$ and a virtual sink $t$; see [134]. The node $\mu$ is connected to all sources, i.e., nodes in $\mathcal{S}$, with a directional edge with infinite capacity; all edges from node $\mu$ to sources are recorded in the set $E_\mu$. Similarly, all sinks in $\mathcal{S}$ are connected with an infinite capacity directional edge to node $t$; the set of directed edges from sinks to node $t$ are stored in $E_t$. Let $G'(V', E')$ denote the revised network, where $V' = V \cup \{\mu\} \cup \{t\}$ and $E' = E \cup E_\mu \cup E_t$. Hence, from here onwards, without loss of generality, maximizing the total flow to one sink is considered, i.e., the virtual sink $t$, from the virtual source $\mu$. It is worth noting that the maximum flow to each sink can be obtained by inspecting the total flow entering each actual sink in the final solution.

3.3 Proposed Solutions

This section proposes three algorithms, namely Path, Tabu and LagOP, to place ACs in a large scale WSN. The Path algorithm aims to recharge all nodes on the shortest path to the sink. It also acts as initial solution in Tabu approach. This is followed by LagOP that utilizes Lagrangian relaxation and sub-gradient optimization.
3.3. Proposed Solutions

3.3.1 Path

This subsection first outlines a heuristic called $Path$ to efficiently determine the sensor nodes in which to park an AC in a large scale WSN. After calling $Path$ to place the ACs, this subsection then sets the corresponding $R_i$ in constraint (3.1) to one, removes constraint (3.5) and solves the resulting LP to obtain the max flow.

The details of $Path$ are shown in Algorithm 1. Let $\hat{R}$ be a set that records the location of all ACs. Recall that there are $R^\#$ ACs. In Line 2, it uses a variable $r$ to denote the total number of unassigned ACs; it is initialized with $R^\#$. Next, in Line 3, $Path$ calls the function $Yen()$ to obtain up to $|S|$ shortest paths, in increasing length order, from the virtual source node $\mu$ to the virtual sink $t$; it uses $P$ to store the paths. The term $P[k]$ indexes the path at position $k$, where the shortest path is at index one. The function $Node(P[k], \hat{R})$ in Line 6 returns all nodes on path $P[k]$ that are not in $\hat{R}$; let $n$ be a set that stores these nodes. If $|n|$ is less than $r$, $Path$ adds all nodes in $n$ to $\hat{R}$ and reduces the total unassigned ACs by $n$; see Line 9-10. Otherwise, in Line 12, $Path$ calls $LowENodes()$ to obtain $r$ nodes with the lowest energy in $n$; let $m$ be a set that stores these nodes. Line 13 adds all nodes in $m$ to $\hat{R}$. If $r$ becomes zero, $Path$ outputs $\hat{R}$ and terminates.

**Proposition 2.** The time complexity of $Path$ is $O(|S||V'|^3)$.

**Proof.** The most computational expensive part of $Path$ is running Yen’s algorithm, which has a run time complexity of $O(|S||V'|^3)$ [135]. Observe that Line 6 runs for at most $O(|V'|)$ times, and Lines 5-16 run at most $O(R^\#|V'|)$ times. In other words, Lines 5-16 have a lower run time complexity than Yen’s algorithm, which proves the proposition.

3.3.2 Tabu

Tabu search is a meta-heuristic approach originally proposed by Glover [136] and aims to obtain a global optimum iteratively. It starts from an initial solution obtained via a heuristic algorithm; in our case, the $Path$ algorithm. It then searches
3.3. Proposed Solutions

Algorithm 1: Path Algorithm

\textbf{Input:} \( G'(V', E') \)  
\textbf{Output:} \( \hat{R} \)

1. \( \hat{R} = \emptyset \);
2. \( r = R# \);
3. \( \mathcal{P} = Yen(G'(V', E')) \);
4. \( k = 1 \);
5. \textbf{while} \( r \neq 0 \) \textbf{do}
   6. \( n = Node(\mathcal{P}[k], \hat{R}) \);
   7. \( k++ \);
   8. \textbf{if} \( |n| \leq r \) \textbf{then}
      9. \( \hat{R}.add(n) \);
      10. \( r = r - |n| \);
   11. \textbf{else}
      12. \( m = LowENodes(r, n) \);
      13. \( \hat{R}.add(m) \);
      14. \( r = 0 \);
   15. \textbf{end}
6. \textbf{end}

the “neighbor” of the current solution to find a local optimal solution. Later, it “moves” to this neighbor and repeats the process. In order to prevent cycling, \textit{Tabu} search uses a short-term memory called a tabu list, denoted by \( TL \), to record visited solutions in the previous \(|TL|\) iterations. This list operates in a First In First Out (FIFO) manner where the oldest solution is removed after \(|TL|\) iterations. \textit{Tabu} search has three termination criteria: (i) the current solution has not improved for a given number of iterations, or (ii) the neighborhood of the current solution is empty, or (iii) \textit{Tabu} search completed a given number of iterations.

Based on \textit{Tabu} search, this subsection presents \textit{Tabu} algorithm that contains the following key steps; see Algorithm 2. First, it initializes \( TL \). It then calls the function PlaceACs() to place all ACs. This serves as the initial solution. Any heuristic methods can be used; in this chapter, \textit{Path} algorithm is used to obtain an initial deployment of ACs. Let \( f \) denote the current max flow using deployment \( \hat{R} \); see Line 2. As there are \( \binom{|V-S|}{R#} \) possible AC deployments, searching every possible solutions results in a high computation overhead. To this end, \textit{Tabu} only searches in the neighborhood of the current solution. Specifically, the neighbors of the current
3.3. Proposed Solutions

solution \( \hat{R} \) is defined as follows,

\[
\mathcal{N}(\hat{R}) = \{(s, t) \mid s \in \hat{R}, t \in V - \mathcal{S} - \hat{R}, (s, t) \notin TL\}
\] (3.6)

In words, the set \( \mathcal{N}(\hat{R}) \) represents all pairs where \( s \in \hat{R} \) and \( t \in V - \mathcal{S} - \hat{R} \). Thus, for each solution, there are up to \( R^\#(|V - \mathcal{S}| - R^\#) \) neighbor solutions.

In Algorithm 2, Line 4 constructs \( \mathcal{N}(\hat{R}) \) as per constraint (3.6). Line 5 then calls the function \( \text{CalculateMaxFlow()} \) to calculate the max flow for each neighbor in \( \mathcal{N}(\hat{R}) \) and returns the one with the maximum value, denoted by \( f^\ast \), and also the corresponding pair \((s^\ast, t^\ast)\) that yielded the the said max value. Note that the function \( \text{CalculateMaxFlow()} \) calls a LP solver with the constraint (3.1), (3.2) and (3.3) to obtain the maximum flow. Line 6 “moves” to this neighbor whereby it adds node \( t^\ast \) to \( \hat{R} \) and deletes node \( s^\ast \) from \( \hat{R} \). Furthermore, in Lines 7-8, the tabu list, i.e., \( TL \), is updated; recall that the first inserted item is removed after \( |TL| \) iterations. Next, \( Tabu \) updates the current max flow value if \( f^\ast \) is larger than \( f \); see Lines 9-11. In addition, in order to check the termination conditions, \( Tabu \) uses the variable \( m \) to record the times in which there is no improvement in max flow value. Thus, if there is no change in a given iteration, the variable \( m \) increases by one; see Line 13. Otherwise, Line 11 resets \( m \) to zero. Let \( MAX_1 \) denote the maximum total number of iterations, and \( MAX_2 \) denote the maximum number of iterations without improvement in max flow; i.e., the bound on the variable \( m \). The function \( Finish(m, MAX_2, \mathcal{N}, MAX_1) \) returns true if \( m \) reaches \( MAX_2 \) or Lines 3-15 have repeated \( MAX_1 \) times. It also returns true if \( \mathcal{N} \) is empty. The output of \( Tabu \) is the max flow \( f \).

**Proposition 3.** The time complexity of \( Tabu \) is \( \mathcal{O}(|V - \mathcal{S}|^5) \).

**Proof.** The most expensive part of \( Tabu \) is incurred by \( \text{ConstructNeighbor}(\hat{R}) \) and \( \text{CalculateMaxFlow()} \). The former function has a time complexity that corresponds to the number of neighbors of a given \( \hat{R} \); i.e., \( \mathcal{O}(R^\#(|V - \mathcal{S}| - R^\#)) \). To compute the max flow of a given topology with \( R^\# \) deployed ACs takes \( \mathcal{O}(|V - \mathcal{S}| \cdot |E'|) \) using
3.3. Proposed Solutions

Algorithm 2: Tabu search

Input: $G'(V', E')$ and $S$

Output: Final max flow $f$

1 $TL = \{\}$;
2 $f = \hat{R} = PlaceACs(G'(V', E'), S)$;
3 repeat
4 $N(\hat{R}) = ConstructNeighbor(\hat{R})$;
5 $[f^*, (s^*, t^*)] = CalculateMaxFlow(N(\hat{R}))$;
6 $\hat{R} = \hat{R} \cup \{t^*\} \setminus \{s^*\}$;
7 $TL.add((s^*, t^*))$;
8 $TL.delete()$;
9 if $f^* > f$ then
10 $f = f^*$;
11 $m = 0$;
12 else
13 $m = m + 1$;
14 end
15 until $Finish(m, MAX_2, N, MAX_1)$;

the algorithm in [137]. As there are up to $|\hat{R}|(|V - S| - |\hat{R}|)$ neighbors, there is $O(|V - S||E|R#(|V - S| - R#))$. To obtain the maximum flow value among the computed max flow values take $O(R#(|V - S| - R#) - 1)$ time. Given the foregone fact, $CalculateMaxFlow()$ thus has a run time complexity of $O(|V - S||E|R#(|V - S| - R#)). Thus, there is $O(MAX_1 \times |V - S||E|R#(|V - S| - R#)). In the worst case, $R# = 0.5 \times |V - S|$, meaning Tabu has a run time complexity of $O(|V - S|^5)$.

3.3.3 LagOP

Lagrangian relaxation (LR) moves “complicating” constraints into the objective function and for each such constraint, it attaches a price or multiplier [69]. The revised objective is then “easier” to solve. Consider the following ILP,

\[
\text{MIN} \quad Cx \\
\text{s.t.} \quad Ax \leq b \\
x \in \{0, 1\}
\]
In order to solve this ILP efficiently, the inequality constraint is moved into the objective function. The revised program, called the Lagrangian Lower Bound Program (LLBP), is thus,

\[
\begin{align*}
\text{MIN} & \quad Cx + \lambda(Ax - b) \\
\text{s.t.} & \quad x \in \{0, 1\}
\end{align*}
\]

The Lagrange multiplier \( \lambda \) is a positive real value. Solving LLBP yields the lower bound for the original ILP; LLBP and the ILP have the same objective value when they are both optimal. To maximize LLBP, the following program is formulated, also called the Lagrangian Dual (LD),

\[
\max_{\lambda \geq 0} \{\text{MIN} \ Cx + \lambda(Ax - b)\} \quad (3.7)
\]

The LD program can then be solved using sub-gradient optimization [69]. The aim is to update the value of \( \lambda \) in a manner that solves the LLBP to yield the minimum lower bound of the original ILP. Specifically, it uses the following equations to adjust the value of \( \lambda \).

\[
\begin{align*}
\lambda &= \max(0, \lambda + \Delta g) \quad (3.8) \\
g &= Ax - b \quad (3.9) \\
\Delta &= \pi \frac{(UB - LB)}{\sum_{1 \leq i \leq |g|} |g_i|^2} \quad (3.10)
\end{align*}
\]

Here \( g \) is the sub-gradient. The term \( \Delta \) denotes the step size. The variable \( \pi \) is used to control the step size with initial value \( \pi_{\text{init}} \). It will reduce if there is no improvement to the lower bound after a given number of iterations. The term \( UB \) and \( LB \) is the upper and lower bound for the ILP respectively. Specifically, the term \( UB \) is obtained by a heuristic algorithm, while \( LB \) is the solution after solving LLBP. If a constraint is violated, then its corresponding multiplier will increase. Lagrangian
relaxation has three termination criteria. First, the expression \( \sum_{1 \leq i \leq |g|} |g_i|^2 \) is zero. That means the optimal value of ILP is equal to the result of LLBP. Second, the term \( \pi \) is smaller than a given value \( \pi_{\text{min}} \). Third, it will terminate after a given number of iterations, denoted by \( \text{MAX}_1 \).

With the aid of Algorithm 3, the approach called LagOP, which uses Lagrangian relaxation and sub-gradient optimization, is introduced. First, the constraints (3.1) and (3.3) are “moved” into the objective function of the MILP and assign multipliers \( \lambda_1^i \) and \( \lambda_{i,u}^2 \) to the two constraints respectively. In particular, to aid presentation, let \( l_1^i \) represent the expression \( R_i B + E_i - \rho \sum_{u \in V_i^-} f_{u,i} - \tau \sum_{v \in V_i^+} f_{i,v} \). The term \( l_{i,u}^2 \) represents the expression \( -f_{i,u} + \sum_{n=1}^N a_{n,i,u} C x_n \). Thus, the following LLBP is obtained,

\[
\text{MIN } \sum_{s \in \delta} \sum_{i \in V_s^-} f_{i,s} + \sum_{i=1}^{|V|-|\delta|} \lambda_1^i l_1^i + \sum_{(i,u) \in E} \lambda_{i,u}^2 l_{i,u}^2 \\
\text{s.t. } (3.2), (3.4), (3.5)
\]

In contrast to ILP, the LLBP of ACP-MF problem yields an upper bound \( UB \). The lower bound of the problem, i.e., \( LB \), are randomly selected and used as the initial value of \( f \), which records the latest best max flow value; see Line 2. LagOP then solves LLBP and calculates the sub-gradient and the step size using (3.8); see Lines 6-7. The value of \( g \) corresponding to a violated constraint is smaller than zero in our problem. Thus, in order to increase the \( \lambda \) of violated constraints, LagOP uses the expression \( \lambda = max(0, \lambda - \Delta g) \) to update \( \lambda \); see Line 8. According to the \( \hat{R} \) calculated by LLBP, LagOP calculates the max flow, denoted by \( f' \). LagOP then updates the max flow \( f \) if \( f' \) is larger than \( f \); see Lines 10-12. If \( f \) is not improved for a given number of iterations \( \text{MAX}_2 \), the function \( \text{NoImprove}(f, f', \text{MAX}_2) \) returns true. The function \( \text{Finish}(\sum_{1 \leq i \leq |g|} |g_i|^2, \pi_{\text{min}}, \text{MAX}_1) \) is used to check the LR termination criteria stated earlier.
Algorithm 3: LagOP

Input: \( G'(V', E') \) and \( S \)
Output: Final max flow \( f \)

1. Initialize \( \lambda \);
2. \( f = LB \);
3. \( \pi = \pi_{\text{init}} \);
4. repeat
5. \( LLBP = \text{ConstructLR}(\lambda) \);
6. \([UB, g, \hat{R}] = \text{SolveProgram}(LLBP)\);
7. \( \Delta = \sum_{1 \leq i \leq |g|} |g_i|^2 \);
8. \( \lambda = \max(0, \lambda - \Delta g) \);
9. \( f' = \text{MaxFlow}(\hat{R}) \);
10. if \( f' > f \) then
11. \( f = f' \);
12. end
13. if \( \text{NoImprove}(f, f', MAX_2) \) then
14. \( \pi = \pi/2 \);
15. end
16. until \( \text{Finish}(\sum_{1 \leq i \leq |g|} |g_i|^2, \pi_{\text{min}}, MAX_1) \);

3.4 Evaluation

Experiments are conducted in Matlab [138] and Matgraph [139]. Parameters originate from the specification of MicaZ [3]. The link capacity is 250 kbps, which corresponds to the data rate of the TI CC2420 transceiver [3]. However, in practice, due to protocol overheads, such as channel contention, the actual data rate is likely to be less than 250 kbps. Hence, the max flow results reported in Section 3.4.1 and 3.4.2 should be interpreted as the theoretical maximum, and correspond to the total maximum flow entering the virtual sink.

Without loss of generality, the method of generating the matrix \( A \) is as follows. First, a vector or transmission set \( Z \) of dimension \( |E| \times 1 \) is constructed. The method then randomly selects a link, say \( l \), and add it to vector \( Z \). All links that conflict, as per the protocol interference model, with \( l \) are removed. The method then selects the next random link. This process is repeated until there are no remaining links. Second, transmission set \( Z \) is checked whether matches any columns in matrix \( A \). If there is no match, \( Z \) is added into \( A \). The generation of matrix \( A \) terminates when
Table 3.2: The value of parameters in the evaluation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$</td>
<td>250 kbps</td>
</tr>
<tr>
<td>$\rho$</td>
<td>209 nJ/b</td>
</tr>
<tr>
<td>$\tau$</td>
<td>226 nJ/b</td>
</tr>
<tr>
<td>$\mathcal{E}_i$</td>
<td>0-15 mW</td>
</tr>
<tr>
<td>$\beta$</td>
<td>30 W</td>
</tr>
<tr>
<td>MAX$_1$ in Tabu</td>
<td>6</td>
</tr>
<tr>
<td>MAX$_2$</td>
<td>5</td>
</tr>
<tr>
<td>$</td>
<td>TL</td>
</tr>
<tr>
<td>MAX$_1$ in LagOP</td>
<td>60</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>0.005</td>
</tr>
</tbody>
</table>

each row in $A$ has at least one entry with a value of one.

The TI CC2420 transceiver consumes 209 and 226 nJ/b for receiving and transmitting one bit, respectively. Therefore, a node consumes 435 nJ to forward one bit.

The energy consumption of the processor is neglected because processing each bit only consumes $4.3 \times 10^{-12}$ Joules; see [140]. Note, if additional energy cost is added, then our results will simply be scaled linearly. Each sensor node is equipped with an Enocean ECS310 solar cell [141]. As per [16], its recharging rate is at most 15 mW in direct sunlight. Further, assume the solar recharging rate is zero at night.

The AC has WPT capability. Its charging rate is up to 50W with 60% charging efficiency [142]. The theoretical link capacity is 250 kbps. In Tabu, $MAX_1$, $MAX_2$ and $|TL|_1$ is set to 6, 5 and 3, respectively. In LagOP, $MAX_1$ is 60. The initial value of $\lambda$ is 0.005. As per [143], $MAX_2$ and $\pi_{min}$ is 30 and 0.005, respectively. Table 3.2 lists the value of parameters used in our evaluation.

The evaluation is first carried out on small networks due to the tractability of MILP. After that, the performance of Tabu, LagOP and Path is evaluated in large scale networks. In addition, let LOWER and UPPER represent the lower and upper bound of the total achievable max flow. The value of UPPER is the max flow when all nodes have an AC; i.e., when $R^# = |V|$. Conversely, LOWER denotes the maximum total flow when no nodes have an AC. In both cases, the attainable max flow is dictated by the well-known max-flow-min-cut theorem.
3.4. Evaluation

3.4.1 Results for Small Networks

This subsection studies the impact of four parameters: number of nodes $|V|$, number of ACs $R^\#$, node degree $\delta$ and number of sources $|S|$. For each experiment, one parameter varies whilst the others fixed. Each algorithm runs 100 times. The location of sources and the sink is chosen randomly in each run.

The first experiment studies the following $\delta$ values: 3, 4, 5, 6 and 7. The value of parameters $R^\#$, $|S|$ and $|V|$ is 3, 3 and 30, respectively. Figure 3.2 shows that increasing $\delta$ has a positive impact on the max flow. For example, when $\delta$ increases from three to seven, the max flow computed by MILP increases as much as 66.08 kb/s; i.e., from 149.66 to 215.74 kb/s. On average, the max flow of Tabu, LagOP and Path is 99.76%, 94.32% and 97.43% that of MILP, respectively. This is because of the following reasons. First, if there is only one route from a source to the sink, the flow is limited by the node with the minimum energy. In contrast, as $\delta$ rises, a source has more neighbors such that the number of routes from sources to the sink increases and thus more data can be forwarded. Moreover, as per constraint (3.1), the available energy of a node determines the amount of data it can forward. An intermediate node may not exhaust its energy when $\delta$ is low. If this intermediate node has a higher node degree, it uses any remaining energy to forward data via other routes. Second, as $|E|$ increases and $|V|$ is fixed, the number of intermediate nodes on paths from a node to the sink decreases. For example, there is a WSN with six nodes. If the node degree is two, the number of intermediate nodes between any two nodes may be zero, one or two. However, if the node degree increases to three, all nodes become neighbors of one another. Consequently, the link between a source and the sink is allocated more active time to increase the max flow. Another observation is the merits of upgrading $R^\# = 3$ nodes among $|V| = 30$ possible nodes. Specifically, the upgrade increases the flow rate for the case without ACs up to 153.86 kb/s, and no more than 37.32% off from $UPPER$. In addition, note that the gap between LagOP and MILP decreases from 12.63% to 2.36% when $\delta$ increases.
from three to seven. This is because when $\delta$ increases, the number of Lagrange multipliers is fixed but there are more decision variables that have influence on the value of each Lagrange multiplier; see Section 3.3.3.

![Figure 3.2: Max flow with varying $\delta$ values.](image)

The second experiment investigates the following values of $|V|$: 10, 30, 50, 70 and 90. The value of $R^\#$, $|S|$ and $\delta$ are set to three. Referring to Figure 3.3, when $|V|$ increases from 10 to 90, the max flow of MILP drops from 216.73 to 106.32 kb/s, or a reduction of 51.85%. On average, the max flow of Tabu, LagOP and Path is 99.66%, 83.69% and 96.70% that of the MILP. The reasons are as follows. As there are only three ACs, this means only three nodes will have a higher energy to forward more data from their neighbors. However, as $|V|$ increases, there are more intermediate nodes between sources and the sink; i.e., the sources have a longer path to the sink. Thus, the recharged nodes have little influence on the final max flow. This is especially significant with increasing $|V|$ as there are more bottleneck nodes. Another observation is that the maximum flow when using MILP increases from 104.36 to 106.32 kb/s as $|V|$ increases from 70 to 90. The reason is as follows. Increasing $|V|$ means there are more links in a WSN. As $\delta$ is fixed, the number
of links that have interference with an active link is fixed. Hence, there are more active links when $|V|$ increases. In addition, as shown in Figure 3.3, the solutions, e.g., Tabu, are able to effectively select three, among 90, nodes to upgrade, which cause the flow rate to increase by $\frac{75.41}{30.21} = 249.62\%$. In addition, note that the maximum gap between $LagOP$ and MILP increases from 0.38\% to 26.45\% when $|V|$ increases from 10 to 90. The reason is as follows. As per Section 3.3.3, as $|V|$ increases, the number of Lagrange multipliers rises. Thus, the difference between $LagOP$ and MILP increases.

Figure 3.3: Max flow with varying $|V|$.

The next experiment investigates the effect of $R^\#$ where it takes on the following values: 1, 3, 5, 7 and 9. Similar to the last experiment, $\delta$ and $|S|$ are three. Also, $|V|$ has a value of 30. Referring to Figure 3.4, when $R^\#$ increases from one to nine with an interval of two, the max flow of MILP increases from 86.69 to 197.00 kb/s. This is an increase of 127.25\%. On average, the max flow of Tabu, $LagOP$ and Path is 98.34\%, 90.40\% and 94.84\% that of MILP, respectively. As expected, increasing $R^\#$ leads to more nodes with a higher energy, which helps increase max flow. In addition, as the topology is fixed, increasing $R^\#$ means more nodes on a path are
likely to be recharged, which leads to a higher flow rate. Figure 3.4 also shows the advantages of upgrading nodes. In particular, compared to LOWER, the max flow increases by 187.82% when $R^\# = 1$. This is because nodes with unused energy on a data transmission path can increase their flow rate when bottleneck nodes on the path are augmented with an AC.

The fourth experiment studies the effect of $|S|$ with the following values: 1, 3, 5, 7 and 9. Parameters $\delta$, $R^\#$ and $|V|$ are set to 3, 3 and 30, respectively. Referring to Figure 3.5, higher $|S|$ values cause the max flow of MILP to increase from 95.45 to 227.47 kb/s. On average, the max flow of Tabu, LagOP and Path is 99.92%, 93.24% and 98.12% that of MILP, respectively. With increasing $|S|$, sources are more likely to be placed on paths with available energy. First consider a scenario with one source, and assume its flow rate is limited by a relay node with a low energy. If $|S|$ increases, a new source may be located between the relay node and the sink. Consequently, the said relay does not affect the flow of the new source. Furthermore, as there are more sources, more data will be generated. As a result, the maximum flow rate increases. In addition, as wireless interference exists, the
sink cannot always receive data from a source in one source scenarios. However, if $S$ increases, the sink may still receive data from other sources when one source has interference with its neighbours. Figure 3.5 also shows the merit of using ACs to increase flow rates. For example, compared to $LOWER$, the flow rate obtained by $Tabu$ increases by 489.90% when $|S| = 9$.

The max flow gap between $Tabu$ and MILP is at most 3.95% when $R\#_R$ is nine. This is because the increase in $R\#$ leads to more possible AC deployments. For example, in a 30 nodes WSN, there are only 30 possible AC deployments when one node is upgraded. By contrast, the number of possible AC deployments increases to 14307150 when there are nine ACs. The maximum gap between $Path$ and MILP is 6.30%. The reasons for the gap are as follows. $Path$ always recharges the nodes on the shortest route from a source to the sink first. However, it does not consider other routes. For example, nodes near the sink that are not on the shortest route also have an influence on the max flow. Furthermore, as the initial energy of each node is generated randomly, recharging the shortest route first may not generate the max flow for a given topology. Finally, if two routes have the same number of nodes, $Path$ may not select the better route.

### 3.4.2 Results for Large Networks

In large networks, there are three experiments to explore the performance of $Tabu$, $LagOP$ and $Path$ in large networks. Specifically, in these large networks, there are 150 nodes. Three parameters are studied: number of ACs ($R\#$), node degree ($\delta$) and number of sources ($|S|$). Each experiment consists of 60 runs. The reason is that large parameters in the experiments are not computationally tractable.

The first experiment studies the influence of $R\#$. The value of $\delta$ and $|S|$ is three and 10, respectively. The value of $R\#$ is varied from 10 to 90. Referring to Figure 3.6, $UPPER$ has no significant changes; the recorded max flow ranges from 202.47 to 202.36 kb/s with increasing $R\#$. When $R\#$ increases from 10 to 90, the max flow
of Tabu increases by 7.11%; i.e., from 183.88 to 192.38 kb/s. On average, the max flow of LagOP and Path is 93.50% and 95.32% that of Tabu, respectively. However, notice that the max flow when using Path only achieves 89.97% that of Tabu when \( R^\# \) is 10. This percentage increases to 97.13% when \( R^\# \) is 90. This is because ten ACs may not be enough to recharge all nodes on the shortest path from a source to the sink. As \( R^\# \) increases, there is a higher chance of recharging all nodes on the shortest path from a source to the sink. In addition, the gap between Tabu and Path drops from 18.45 kb/s to 5.69 kb/s when \( R^\# \) increases from 10 to 90. This is because the increase in \( R^\# \) results in more nodes with extra energy to forward data. Figure 3.6 also shows the effectiveness of Tabu. In particular, Tabu is able to upgrade 30 nodes, among 150 nodes, to produce almost equivalent flow rates as UPPER, which requires 150 ACs.

The next experiment studies varying \( \delta \) values from three to seven. Both \( R^\# \) and \( |S| \) are 10. Referring to Figure 3.7, as expected, the increase in \( \delta \) has a positive influence on max flow. For example, when \( \delta \) increases from three to seven, the max flow of UPPER increases by 16.86%; i.e., from 213.93 to 250 kb/s. Another
observation is that the max flow of Tabu, LagOP and Path is 202.68, 194.69 and 192.82 kb/s respectively when $\delta$ is three. As $\delta$ rises to seven, the max flow of Tabu, LagOP and Path increases by 22.31%, 23.20% and 24.22% respectively. On average, the max flow of LagOP and Path is 94.45% and 93.77% that of Tabu, respectively.

The third experiment increases $|S|$ from 10 to 90. The value of $R^\#$ and $\delta$ is ten and three, respectively. Referring to Figure 3.8, the max flow of UPPER increases from 213.93 to 245.83 kb/s when $|S|$ increases from 10 to 90. Note, the max flow of UPPER is same as that of Tabu when $|S|$ is 90. However, Tabu only use ten ACs as opposed to 150 ACs for UPPER. This is because Tabu can simply deploy an AC next to the nodes that are adjacent to sinks. Sources manage to saturate or use the available energy at these nodes. Further, the max flow attained by LagOP and Path is 96.06% and 95.13% that of Tabu in scenarios with 10 sources. As $|S|$ rises to 90, the max flow of LagOP and Path is 99.64% and 99.78% that of Tabu respectively. Note that the performance of Path is close to that of Tabu when $|S|$ increases. The reason is as follows. The max flow rate derived by Tabu is equal to UPPER when $|S|$ is 90. As for Path and increasing $S$, the number of nodes on the shortest path
from sources to the sink decreases, meaning \textit{Path} can upgrade more paths. Thus, the gap between \textit{Tabu} and \textit{Path} decreases when $S$ increases.

Next, the run time of all algorithms with varying $|S|$ values is plotted; see Figure 3.9. The increase of $|S|$ does not have a significant influence on the run time of \textit{Tabu}. This is because only $R^\#$ and $|V|$, which determine the number of neighbours solutions in each iteration, have an impact on the run time of \textit{Tabu}; see (3.6). Another observation is that the running time of \textit{LagOP} varies from 1.58 to 1.75 seconds. This is because $|S|$ has no impact on the number of constraints in the LLBP of our problem. Next, the run time of \textit{Path} is considered. According to Figure 3.9, the run time of \textit{Path} is 1.79 seconds when $|S|$ is 10. As $|S|$ rises to 90, the run time of \textit{Path} increases by 930.17%. This is because the time complexity of \textit{Path} is related to $|S|$; see Proposition 2. On average, the run time of \textit{Tabu}, \textit{LagOP} and \textit{Path} is 58.44, 1.65 and 10.48 seconds, respectively. The impact of parameters $R^\#$ and $\delta$ are also investigated. As they have no discernible impact on all tested algorithms, they are omitted from this chapter.

Lastly, this chapter compares flow rate fairness. To do this, Jain’s fairness index
3.4. Evaluation

Figure 3.8: Max flow with varying $|S|$ values in large networks.

Figure 3.9: Impact of $|S|$ on running time in large networks.
is employed. The value of $R^\#$, $|S|$ and $\delta$ are set to three. The value of $|V|$ increases from 10 to 90 with an interval of 10. Next, MILP, LagOP, Path and Tabu runs 100 times for each value of $|V|$. Referring to Figure 3.10, the fairness value of MILP, Tabu, Path and LagOP is 0.40, 0.41, 0.39 and 0.38 respectively when $|V|$ is 10. As $|V|$ rises to 90, the fairness value of MILP, LagOP, Path and Tabu increases by 72.74%, 60.99%, 54.28% and 95.51%, respectively. This is because the following reasons. The number of intermediate nodes between sources to the sinks decreases as $|V|$ increases. An AC has a higher influence on the data rate of sources near the sink in scenarios with small number of nodes. For example, in 10 nodes scenarios, a source may be next to the sink. In order to maximize the flow at the sink, one method is that deploys an AC next to the source and allocates all active time to the link between the source and the sink. Consequently, the remaining sources cannot transfer data to the sink. On average, MILP, LagOP, Path and Tabu achieve a fairness value of 0.53, 0.60, 0.48 and 0.54, respectively. When using Path, sources have the worst fairness as compared to the other three algorithms. This is because Path upgrades either all nodes on the shortest path or $R^\#$ nodes. Hence, only the sources that transfer data via upgraded nodes can increase their flow rate.

Table 3.3 summarizes the obtained results. LagOP, Path and Tabu attains 91.65%, 97.00% and 99.40% of MILP in small networks, respectively. In large network, Tabu has the best performance but the highest running time. The max flow rate of LagOP is almost equal to that of Path. However, the running time of Tabu is approximately six times longer than that of LagOP.

<table>
<thead>
<tr>
<th>Network Size</th>
<th>Algorithm</th>
<th>Max Flow Rate (kb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small Network</td>
<td>MILP</td>
<td>167.93</td>
</tr>
<tr>
<td></td>
<td>LagOP</td>
<td>153.91</td>
</tr>
<tr>
<td></td>
<td>Path</td>
<td>162.89</td>
</tr>
<tr>
<td></td>
<td>Tabu</td>
<td>166.93</td>
</tr>
<tr>
<td>Large Network</td>
<td>LagOP</td>
<td>208.02</td>
</tr>
<tr>
<td></td>
<td>Path</td>
<td>208.43</td>
</tr>
<tr>
<td></td>
<td>Tabu</td>
<td>217.62</td>
</tr>
</tbody>
</table>
This chapter has investigated the novel problem of upgrading a subset of sensor nodes with the aim of maximizing the flow rate at one or more sinks. The problem is modeled as a MILP. Three novel solutions are proposed to upgrade sensor nodes in large scale WSNs. The results show that the performance of Path and LagOP are close to that of Tabu. Nevertheless, both Path and LagOP have a much smaller running time than Tabu.

As mentioned in Section 2.2.1, throughput maximization results in unfair rate allocation among sources, which reduces sensing quality. In addition, Figure 3.10 also shows that the key problem of ACP-MF is unfair rate allocation. Therefore, the next chapter addresses a similar problem but the objective is fair rate allocation.
ACs Placements and Min Flow Rate

Fair rate allocation is of concern to surveillance and monitoring applications. For example, if a WSN is used to monitor the health of a bridge, see [23], sensor nodes are required to have a high frequency of data collection rate in order to accurately characterize the vibrations caused by crossing vehicles. However, sensing quality is determined by the energy harvesting rate of sensor nodes, which is unpredictable and uncontrollable [16]. For example, solar-equipped sensor nodes may be deployed on the sides of a bridge and are thus not continuously exposed to the sun. Consequently, the sensing and/or relaying rates of those sensor nodes would be less as compared to nodes with more exposure to the sun.

This chapter employs WPT and energy harvesting to address the ACP-MM problem: placing a given number of ACs among all possible nodes to maximize the minimum rate of all sources in an energy harvesting WSN. The problem is modeled as a MILP. In order to solve the problem in large scale WSNs, two greedy algorithms are proposed. The first, called greedy node deployment (GND), iteratively upgrades a node that yields the highest increase in max-min rate. However, a key problem is that it has to search all non-upgraded nodes, and thus incurs a high computational cost. To this end, a second approach, called one unit energy deployment (OUED) algorithm, is proposed. It uses a relaxed version of the MILP to share one unit energy
4.1 Preliminaries

A rechargeable WSN can be modeled as a graph $G(V \cup t, E)$, where $V$ and $E$ denote the set of sensor nodes and directional links, respectively. In addition, $t$ denotes the sink. The WSN contains a set of sources, relay nodes and one sink. The sources, denoted by $S \subseteq V$, generate data and transfer sensed data to a sink via relay nodes. The relay nodes by definition cannot generate data, and only receive and forward data. Let $g_i$ represent the data generation rate of node $i$. If node $i$ is a relay node, $g_i$ is zero. Let $N_i \subseteq V$ represent the neighbors of node $i$. Denote $N_i^- \subseteq N_i$ to be the set containing all neighbors that send data to node $i$. Conversely, the set $N_i^+ \subseteq N_i$ are neighbors that node $i$ transfers data to. Denote $f_{i,j}$ to be the transmission rate, bits/s, from node $i$ to $j$. Conversely, $f_{j,i}$ represents the number of bits node $i$ receives from node $j$ per second. A link $(i,j)$ exists if the Euclidean distance between node $i$ and $j$ is smaller than their transmission range. Each link has capacity $C$.

With regard to wireless interference, matrix $A$ represents transmission sets; each row corresponds to a link and each column represents a transmission or independent set. Specifically, an entry $a^n_{i,j}$ in $A$ denotes whether a link $(i,j)$ is active in column $n$. For example, if $a^n_{i,j} = 1$, link $(i,j)$ is active in the transmission set $n$. The set of links in column $n$ can be determined using either the protocol or physical interference model [133]. That is, each column denotes the set of links that can transmit simultaneously without interfering with one another as per the said model.
The transmission or link schedule is represented by a vector $S = [x_1, x_2, \ldots, x_N]$ where $x_n$ is the active time of column $n$ of matrix $A$ and $\sum_{n=1}^{N} x_n = 1$.

In terms of energy, the sink has ample energy. Other nodes, however, have two energy sources: solar and WPT. In addition, they have a storage for harvested energy. The energy harvesting rate of node $i$ is $E_i$ Joule per second. However, distinct locations have different energy harvesting rates. This means some sensor nodes may require an additional energy source. Specifically, ACP-MM aims to deploy $R\#$ ACs with WPT ability to recharge nodes such that the minimum sensing rate is maximized. Assume that each AC only recharges one node. If an AC is parked next to a node $i$, the energy recharging rate of node $i$, denoted as $R_i$, will increase by $\beta$; i.e., formally, $R_i = E_i + \beta$. Lastly, $\tau$, $\rho$ and $\sigma$ denote, respectively, the energy cost of transmitting, receiving and sensing (in Joule per bit).

### 4.2 Problem Formulation

This section models the ACP-MM problem as a MILP with four constraints. This first is energy constraint, specifically

$$\sigma g_i + \rho \sum_{u \in N_i^-} f_{u,i} + \tau \sum_{v \in N_i^+} f_{i,v} \leq R_i \beta + E_i, \forall i \in V \tag{4.1}$$

The LHS represents the energy consumption rate of node $i$. The RHS corresponds to a node’s recharging energy rate. Thus, the energy consumption rate of a node must be no more than its recharging energy rate. Note that a binary variable $R_i \in \{0, 1\}$ denotes whether a node $i$ is recharged by an AC. For example, if $R_i$ is one, an AC is parked next to node $i$. Flow conservation at each node is represented by constraint (4.2), where the output flow at node $i$ is equal to the sum of the input flow and total generated data of node $i$.

$$\sum_{u \in N_i^-} f_{u,i} + g_i = \sum_{v \in N_i^+} f_{i,v}, \forall i \in V \tag{4.2}$$
Next, is the constraint on link capacity. Formally,

\[ f_{i,u} \leq \sum_{n=1}^{N} a_{i,u}^{n} x_{n} C, \quad (4.3) \]

Further, the length of superframe is at most one. Thus,

\[ \sum_{n=1}^{N} x_{n} = 1, \quad (4.4) \]

Next, the number of deployed ACs is no more than \( R^\# \). Formally,

\[ \sum_{i \in V} R_{i} \leq R^\#, \forall i \in V \quad (4.5) \]

Lastly, the objective is to maximize the minimum sensing rate of all sources. That is,

\[
\text{MAX} \quad \text{MIN}\{g_{i}\}_{i \in S} \\
\text{subject to (4.1), (4.2), (4.3), (4.4), (4.5)}
\]

The formulated MILP can only be used to obtain a solution for small scale WSNs. This is because the search space grows according to \( \binom{|V|}{R^\#} \). In fact, in Chapter 3, ACP-MF is shown to be NP-hard. Note, ACP-MM is different because its objective is to maximize the min flow rate of all sources as opposed to finding an AC deployment that yields the maximum flow rate at the sink. Moreover, as shown in Section 4.5, Path algorithm leads to very low, and some times zero, increase in max-min rate.

### 4.3 Proposed Solutions

In order to upgrade nodes in large scale WSNs, two algorithms are proposed: GND and OUED. Both GND and OUED have \( R^\# \) stages. GND algorithm is a greedy
4.3. Proposed Solutions

algorithm that iteratively parks an AC next to a node yielding the highest increase in max-min rate. OUED first relaxes the integer constraint of the formulated MILP and replaces $\beta$ by one unit of energy. The resulting LP is then used in each iteration to determine each node’s share of the one unit energy. After that, OUED identifies the sensor node with the highest share in a given stage and assigns it an AC.

4.3.1 Greedy Node Deployment (GND) Algorithm

GND contains $R^\#$ stages. For each stage, GND selects a node from the set of non-upgraded nodes and deploys an AC next to it. The selected node must yield the highest increase in max-min rate. However, if there are several nodes that produce the same max-min rate, GND randomly upgrades one of them. Further, if there is no node that can be upgraded to increase the max-min rate, GND will select an non-upgraded node randomly. GND then repeats the same process for the next AC until it deploys all $R^\#$ ACs.

The details of GND are shown in Algorithm 4. Let $\hat{R}$ be a set that records the location of each AC. The set $\mathcal{V}$ records all non-upgraded nodes. In each stage, Lines 3-13 deploy an AC. Specifically, Lines 4-8 iterate through all nodes in the set $\mathcal{V}$. In particular, Line 5 parks an AC next to node $j$; consequently, $E_j$ increases by $\beta$. Line 6 calls $\text{runLP()}$ to calculate the max-min rate $F_j$ when an AC is deployed next to node $j$. Line 7 removes the AC from node $j$ to upgrade the next node. Line 9 calls $\text{maxLocation()}$ to return a node $n^*$ with the highest max-min rate in $F$. GND then permanently places an AC next to node $n^*$; see Line 10. After that Line 11 adds node $n^*$ into $\hat{R}$. Lastly, Line 12 deletes node $n^*$ from $\mathcal{V}$; i.e., GND no longer considers it in subsequent iterations.

Now consider applying GND on the WSN shown in Figure 1.4. Assume that there are three ACs. When an AC is parked next to a node, the energy harvesting of this node increases, which upgrades its relay capacity by additional 10 pkt/s. In the first stage, it upgrades node $C$ such that the max-min rate increases to 4 pkt/s.
4.3. Proposed Solutions

Algorithm 4: GND Algorithm

<table>
<thead>
<tr>
<th>Algorithm 4: GND Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $G(V,E)$, $R^#$, $E$</td>
</tr>
<tr>
<td><strong>Output:</strong> $\hat{R}$</td>
</tr>
<tr>
<td>1 $\hat{R} \leftarrow \emptyset$;</td>
</tr>
<tr>
<td>2 $V \leftarrow V$;</td>
</tr>
<tr>
<td>3 for $i \leftarrow 1$ to $R^#$ do</td>
</tr>
<tr>
<td>4 for $j \leftarrow 1$ to $</td>
</tr>
<tr>
<td>5 $\mathcal{E}_j \leftarrow \mathcal{E}_j + \beta$;</td>
</tr>
<tr>
<td>6 $F_j \leftarrow \text{runLP} \left( \mathcal{E}_j \right)$;</td>
</tr>
<tr>
<td>7 $\mathcal{E}_j \leftarrow \mathcal{E}_j - \beta$;</td>
</tr>
<tr>
<td>8 end</td>
</tr>
<tr>
<td>9 $n^* \leftarrow \text{maxLocation} \left( F \right)$;</td>
</tr>
<tr>
<td>10 $\mathcal{E}<em>{n^*} \leftarrow \mathcal{E}</em>{n^*} + \beta$;</td>
</tr>
<tr>
<td>11 $\hat{R}.add \left( n^* \right)$;</td>
</tr>
<tr>
<td>12 $V.delete \left( n^* \right)$;</td>
</tr>
<tr>
<td>13 end</td>
</tr>
</tbody>
</table>

However, in the second stage, as GND can park only one AC at a time, it fails to find one node in $V = \{A, B, D, E, F\}$ that can be upgraded to increase the max-min rate. Thus, in this stage, GND upgrades a node randomly. Assuming GND upgrades node $D$ ($E$) in the second stage, it would be able to upgrade node $E$ ($D$) in the third stage, and thus the algorithm could increase the max-min rate to 8 pkt/s. However, if in the second stage GND upgrades either node $A$, $B$ or $F$, the max-min rate remains at 4 pkt/s because, in stage 3, parking the third AC at any one of the remaining non-upgraded nodes would not improve the max min rate.

### 4.3.2 One Unit Energy Deployment (OUED) Algorithm

GND aims to obtain the highest max-min rate increase in each stage. However, it runs LP many times in each stage; see Proposition 4. This is computationally expensive. Thus, this subsection presents a new algorithm called OUED that contains $R^\#$ stages and only runs LP one time in each stage. In particular, for each stage, OUED shares one unit of energy among all nodes and upgrades the node with the highest share. In order to obtain the share of the one unit energy for each node, the MILP is relaxed to an LP as follows. First, $\beta$ is replaced by one unit of energy.
Next, $R_i$ is replaced by $R'_i$, where $R'_i \in [0,1], \forall i \in V$. In other words, OUED relaxes the integer constraint of the MILP. Thus, there are following constraints:

\begin{equation}
\sigma g_i + \rho \sum_{u \in N^-_i} f_{u,i} + \tau \sum_{v \in N^+_i} f_{i,v} \leq R'_i + \mathcal{E}_i, \forall i \in V \tag{4.6}
\end{equation}

\begin{equation}
\sum_{i \in V} R'_i = 1, \forall i \in V \tag{4.7}
\end{equation}

Next, constraint (4.6) and (4.7) replace constraint (4.1) and (4.5), respectively. Consequently, the LP aims to determine how the one unit of energy is shared among nodes to achieve max-min rate. The value of $R'_i$ represents the share of the one unit energy at node $i$. Next, OUED upgrades the node with the highest share. When there are multiple nodes with the same highest share, OUED randomly upgrades one of the nodes. OUED repeats the same process for the next AC until it deploys all $R^\# ACs$.

Algorithm 5 shows the details of OUED. Lines 2-7 present the process of deploying one AC in a given stage. In particular, OUED calls $relaxLP()$ to compute $R'_i$ for each node $i$; see Line 3. Note that these values are fractional. Line 4 calls $argmax()$ to obtain the node $n^*$ that corresponds to the highest value in $R'$. Line 5 increases the recharging rate of node $n^*$ by $\beta$. That means OUED deploys an AC next to node $n^*$. Finally, Line 6 adds node $n^*$ into $\hat{R}$.

**Algorithm 5: OUED Algorithm**

```
Input: $G(V, E), R^\#, \mathcal{E}$
Output: $\hat{R}$
1 $\hat{R} \leftarrow \emptyset$;
2 for $i \leftarrow 1$ to $R^\#$ do
3     $R' \leftarrow relaxLP()$;
4     $n^* \leftarrow arg\max_{j \in V} R'_j$;
5     $\mathcal{E}_{n^*} \leftarrow \mathcal{E}_{n^*} + \beta$;
6     $\hat{R}.add(n^*)$;
7 end
```
Consider applying OUED for the WSN in Figure 1.4. The value of $R^\#$ and $\beta$ are same as that in the previous example. First, OUED runs an LP solver to allocate one unit of energy with constraints (4.6) and (4.7). OUED allocates this one unit energy to node $C$. Thus, the algorithm upgrades node $C$ in the first stage. Next, OUED runs the LP solver again in the second stage, and $D$ and $E$ are allocated half unit of energy. For this stage, OUED upgrades either node $D$ or $E$ randomly, e.g., node $D$. Then, OUED runs the LP solver again in the third stage and allocates one unit energy to node $E$. As a result, OUED upgrades nodes $C$, $D$ and $E$ that increases the max-min rate to 8 pkt/s.

4.4 Analysis

This section outlines several properties of GND and OUED and prove their worst case performance in perfect binary trees. Let $OPT$ be the objective value returned by the MILP.

**Proposition 4.** GND runs the LP-solver $R^\#|V| - \frac{(R^\# - 1)R^\#}{2}$ times.

**Proof.** Consider Algorithm 4. The LP-solver is called $|V| - i + 1$ times in each $i$ stage. In addition, Lines 4-8 are called $R^\#$ times or equivalently,

$$|V| + (|V| - 1) + \cdots + (|V| - R^\# + 1) = R^\#|V| - (1 + 2 + \cdots + R^\# - 1) = R^\#|V| - \frac{(R^\# - 1)R^\#}{2}$$

This proves the proposition. \hfill \Box

**Proposition 5.** OUED runs the LP-solver $R^\#$ times.

**Proof.** In Algorithm 5, Lines 2-7 repeat $R^\#$ times. Line 3 runs the LP-solver one time in each stage. Thus, OUED runs the LP-solver for a total of $R^\#$ times. \hfill \Box
Proposition 6. For a given WSN $G(V, E)$ with $|S|$ sources, the formulated MILP has $|S| + 2|E| + |V|$ decision variables.

Proof. The decision variables are $g_i$, $f_{u,i}$, $R_i$ and $x_n$. The value of $g_i$ is equal to the number of sources; i.e., $|S|$. In terms of $f_{u,i}$, there are $|E|$ flows. Further, as each node has a decision variable $R_i$, the number of $R_i$ is $|V|$. In addition, each column of matrix $A$ has a decision variable to determine its active time. Thus, the number of $x_n$ is $N$. Consequently, the MILP has $|S| + |E| + |V| + N$ decision variables. □

Proposition 7. For a given WSN $G(V, E)$ with $|S|$ sources, the formulated MILP has $2|V| + |E| + 2$ constraints.

Proof. First, all $|V|$ nodes have an energy and flow conservation constraint. Next, each link is bounded by its capacity. Lastly, there is only one constraint of type (4.4) and (4.5). In total, we have $2|V| + |E| + 2$ constraints. □

As shown in Proposition 4 and 5, $GND$ significantly runs slower than $OUED$ because it runs LP-solver $|V| - (R# - 1)/2$ times more than $OUED$. Further, following Proposition 6 and 7, the complexity of MILP escalates when the size of WSN increases, and thus the MILP solution is not recommended for use in large sized WSN. The simulation in Section 4.5 supports these analyses.

The following proposition sets the upper bound on max-min rate of a WSN that has a sink with $\delta$ node degree and sources with the maximum harvesting rate of $E_{max}$.

Proposition 8. For any topology with one sink and $|S|$ sources, its max-min rate is bounded by $MIN\left\{ \frac{E_{max}}{\sigma + \tau}, \frac{C}{|S|} \right\}$.

Proof. Recall that $\sigma$, $\tau$ represent the energy consumption for sensing and transmitting a bit, respectively. The theoretical capacity is $C$. Referring to constraint (4.1), the sensing rate of source $i$ is restricted by its energy harvesting rate. Let $E_{max}$ be the maximum energy harvesting rate of sources. Thus, for a source, its maximum sensing rate is $\frac{E_{max}}{\sigma + \tau}$. Next, consider the maximum data rate at the sink. The sink
4.4. Analysis

has \( \delta \) incoming links. However, as the adjacent links of the sink cannot be simultaneously active, the flow rate received by the sink is at most \( C \). Further, by constraint (4.3), the amount of data generated at all sources is equal to the amount of data received by the sink. Thus, the amount of data that all sources generate is at most \( C \). If there are \(|S|\) sources, the max-min rate is at most \( \frac{C}{|S|} \). Therefore, the max-min rate cannot exceed either \( \frac{E_{\max}}{\sigma + \tau} \) or \( \frac{C}{|S|} \), meaning it is bounded by \( \min\{\frac{E_{\max}}{\sigma + \tau}, \frac{C}{|S|}\} \), which proves the proposition.

The remaining Propositions 9-12 also assume a perfect binary tree with \( h + 1 \) levels. The root is located at level \( h = 0 \) and all leaf nodes are sources. In addition, all sensor nodes have the same energy harvesting rate of \( E \) Joule per second. Let \( \frac{\text{ALGO}}{\text{OPT}} \) be the ratio of the max-min rate obtained by an algorithm ALGO over MILP.

**Proposition 9.** A sensor node at level \( i \) afford only up to \( \frac{2^i E}{2^i (\rho + \tau)} \) of its energy to forward data from each descendant source.

**Proof.** Let \( s_i \) be the number of descendant sources of a node located at level \( i \). Specifically, \( s_i = 2^{h-i} \). Observe that a node at level \( i \) forwards all data from its descendants. Thus, each descendant has the following share of the node’s energy harvesting rate,

\[
\frac{E}{s_i(\rho + \tau)} = \frac{E}{(\rho + \tau)2^{h-i}} = \frac{2^i E}{2^h(\rho + \tau)}
\]

where \( \rho + \tau \) corresponds to the cost of forwarding.

Note that Proposition 9 equivalently means that each node on level \( i \) can afford only up to \( \frac{2^i}{2^h} \) fraction of its energy rate, i.e., share, to forward data from each source. Further, one can observe that the smallest share is afforded by nodes at level \( i = 1 \); i.e., the children of the sink/root allocates the smallest share to sources/leaves, meaning they are the bottleneck nodes and thus must be upgraded first to improve the max-min rate. Consequently, in order to produce the optimal max-min rate,
nodes must be upgraded in increasing order of levels; i.e., nodes on level \(i + 1\) are upgraded only after all nodes on level \(i\) have been upgraded, starting from \(i = 1\).

**Proposition 10.** The upper bound of \(\text{OPT}\) is \(\frac{\mathcal{E}(R^# + 2)}{2^{i(\rho + \tau)}}\).

*Proof.* As nodes at level one are the bottlenecks, see Proposition 9, MILP will deploy ACs from these nodes onwards. Note that in order to increase the max-min rate, all bottleneck nodes at the same level must be upgraded. Let \(x\) denote the level in which MILP fails to upgrade all nodes given \(R^#\) ACs. As there are \(2^i\) nodes at level \(i\), the number of nodes at level one to \(x - 1\) and \(x\) is respectively \(2^1 + 2^2 + \cdots + 2^{x-1}\) and \(2^1 + 2^2 + \cdots + 2^x\). Hence,

\[
2^1 + 2^2 + \cdots + 2^{x-1} \leq R^# < 2^1 + \cdots + 2^{x-1} + 2^x \tag{4.9}
\]

\[
2^x - 2 \leq R^# < 2^{x+1} - 2 \tag{4.10}
\]

Solving for \(x\), there is \(\log_2(R^# + 2) - 1 < x \leq \log_2(R^# + 2)\). As \(x\) is bounded by \(\log_2(R^# + 2)\), the smallest share after applying MILP is at most \(\frac{\mathcal{E}(R^# + 2)}{2^{i(\rho + \tau)}}\), as required.

**Proposition 11.** The ratio \(\frac{\text{GND}}{\text{OPT}}\) is lower bounded by \(\frac{2}{R^# + 2}\).

*Proof.* Consider \(\text{GND};\) see Algorithm 4. It will randomly upgrade a node if it fails to increase the max-min rate; see Line 9. At level one, there are two bottleneck nodes and \(R^#\) ACs. However, \(\text{GND}\) only upgrades one node each time. Thus, in the worst case, \(\text{GND}\) fails to upgrade the two nodes at level one because available ACs are deployed next to nodes located on other levels. Hence, in the worst case, using Proposition 9 with \(i = 1\), \(\text{GND}\) yields a minimum share of \(\frac{2\mathcal{E}}{2^{i(\rho + \tau)}}\), and applying Proposition 10, \(\frac{\text{GND}}{\text{OPT}} = \frac{2}{R^# + 2}\).

**Proposition 12.** OUED produces the optimal max-min rate, i.e., \(\frac{\text{OUED}}{\text{OPT}} = 1\), for perfect binary trees.

*Proof.* Recall that \(\text{OUED}\) uses an LP to divide the one unit of energy in each iteration. It then upgrades the node with the highest share; see Lines 4-5. If
multiple nodes have the same share, then \textit{OUED} randomly upgrades one of them. \textit{OUED} is shown that always assigns a larger share to a bottleneck node starting at level $i = 1$. Specifically, it will upgrade all nodes on level $i$ first before moving to nodes on level $i + 1$. In doing so, \textit{OUED} has the same behaviour as MILP, meaning it is optimal.

To prove this fact, consider how the one unit of energy is divided by an LP-solver. In particular, the LP-solver uniformly distributes the energy among all bottleneck nodes. There are two cases to consider: (i) it assigns nodes on level $i$ with a share of $\frac{1}{2^i}$. This case occurs when the LP-solver is unable to allocate energy to nodes on level $i$ such that they afford the same share as nodes on level $i + 1$. This also means nodes on level $i + 1$ do not have any share of the one-unit energy. As an example, consider $i = 3$ and $h = 5$. The nodes on level $i = 3$ will assign $\frac{1}{4}$ to each source. To improve the max-min rate, the LP-solver must assign $\frac{1}{4}$ to all nodes on level $i = 3$ so that they have the same fraction as nodes on level $i = 4$. However, only $\frac{1}{8}$ of the one-unit of energy is available for each node on level $i$, meaning nodes on level $i$ remain the bottlenecks. In this case, \textit{OUED} will randomly upgrade a bottleneck node on level $i$, (ii) one or more nodes on level $i$ are given at least $\frac{2^i + 1}{2h}$ fraction of the one unit energy. In this case, the LP-solver is successful in assigning additional energy to nodes on level $i$ such that they and nodes on level $i + 1$ become bottleneck nodes; i.e., they determine the max-min rate of sources. As an example, let $h = 4$. In this perfect binary tree, the nodes on level $i = 1$ assign $\frac{1}{8}$ to each source. The LP-solver assigns $\frac{1}{8}$ to the two bottleneck nodes on level $i$. Consequently, they afford the same share to sources as those on level $i + 1$. Observe that $\frac{3}{4}$ of the one-unit energy remains, which the LP-solver then distributes to non-upgraded nodes on level $i$ and $i + 1$. Let $\gamma$ be the number of nodes on level $i$ that have received some fraction of the one-unit energy. Observe that the LP-solver will first assign the fraction $\frac{2^i + 1}{2h}$ to nodes on level $i$. After doing so, the $\gamma$ nodes on level $i$ along with those on level $i + 1$ are bottlenecks that determine the max-min rate of sources. The LP then distributes the remaining energy to the $\gamma$ nodes plus all nodes on level $i + 1$; i.e.,
the residual energy is divided uniformly amongst $\gamma + 2^{i+1}$ nodes. Formally, each of these nodes receives the following fraction,

$$\epsilon = \frac{1 - \gamma 2^{i+1}}{\gamma + 2^{i+1}}$$

(4.11)

As the $\gamma$ nodes on level $i$ will also receive an additional $\epsilon$ fraction of energy, a node on level $i$ will always receive a higher fraction than a node on level $i + 1$; thus $OUED$ will upgrade nodes on level $i$ before any node on level $i + 1$. As both (i) and (ii) are true, the claim $\frac{OUED}{OPT} = 1$ follows.

4.5 Evaluation

To evaluate the performance of $GND$ and $OUED$, the experiments are conducted in Matlab [138] with Matgraph [139] and CPLEX [78]. The experiments use the parameters of MicaZ. Specifically, the theoretical link capacity is set to 250 kb/s. According to [132], the value of $\sigma$, $\rho$ and $\tau$ is 150, 300 and 300 nJ/b, respectively. All sensor nodes are equipped with an Enocean ECS310 solar cell [141] with a recharging rate of 150 mW in direct sunlight and 1.5 mW in cloudy days. The experiments use real solar irradiance data retrieved from Southwest Solar Research Park, Phoenix, Arizona, USA [144] on the 16-th of April 2013; the recharging rate is a sinusoidal function peaking at 12 o’clock. Thus, on average, the energy harvesting rate of nodes ranges from 0 to 75 mW. In terms of WPT, the charging rate of an AC is 50 W and has a charging efficiency of 60% [142].

With regard to the generation of the matrix $A$, a transmission set $Z$ of dimension $|E| \times 1$ is constructed. Next, a link $l$ is randomly selected. It is then added to $Z$. Note, all links that are conflict, as per the protocol interference model, with $l$ are removed. The method then selects the next random link. This process is repeated until there are no remaining links. Transmission set $Z$ is checked to determine
whether it matches any columns in matrix $A$. If there is no match, $Z$ is added into $A$. A new $Z$ is then generated. The process stops when each row in $A$ has at least one entry with a value of one; i.e., each link exists in at least one transmission set.

The max-min rate obtained by MILP, GND and OUED is compared against the Path algorithm in Chapter 3. Recall that Path aims to maximize the flow rate at the sink. It always selects all nodes on the shortest path to be recharged. After all ACs are parked, the max-min rate at the sink is recorded.

### 4.5.1 Small Networks

This section studies the effect of four network parameters: number of nodes $|V|$, node degree $\delta$, number of ACs $R^\#$ and number of sources $|S|$. In each experiment, one parameter changes whilst the others are fixed. Further, each experiment is conducted 100 times on arbitrary topologies with sources selected randomly. In addition, let $\text{LOWER}$ and $\text{UPPER}$ represent the lower and upper bound of the max-min rate. Specifically, $\text{UPPER}$ is the max-min rate when each node is upgraded. In contrast, $\text{LOWER}$ is the max-min rate when $R^\# = 0$; i.e., no sensor nodes are upgraded.

The first experiment studies the impact of $|V|$ with the following values: 10, 30, 50, 70 and 90. Referring to Figure 4.1, the max-min rate of MILP, GND and OUED decreases with the increasing $|V|$. Specifically, the max-min rate of MILP decreases from 64.39 to 55.96 kb/s when $|V|$ increases from 10 to 90. The max-min rate of GND and OUED drops by 12.25% and 12.28% from 63.44 and 63.28 kb/s, respectively. In addition, $\text{UPPER}$ and $\text{LOWER}$ also reduces as much as 3.00 and 10.84 kb/s from 67.36 and 33.34 kb/s, respectively. This is because increasing $|V|$ results in more relay or intermediate nodes. This means the number of ACs is not sufficient to upgrade these relay nodes. Another observation is that $\text{UPPER}$ increases 4.36 kb/s when $|V|$ increases from 70 to 90. The reason is because there are more active lines when there are more nodes; i.e., higher $|V|$ values. Note that the max-min rate of PATH stays at 19.41 kb/s when $|V|$ increases from 30 to 90.
However, when $|V|$ is ten, the max-min rate of $PATH$ is only 4.11 kb/s. The reason is as follows. When there are only ten nodes, the number of intermediate nodes between a source and the sink is low. Thus, when using $PATH$, three ACs may be sufficient to upgrade all nodes on the shortest path from the source to the sink. Thus, the nodes on the path have ample energy. In order to maximize the flow rate at the sink, all active time can be allocated to the links on the path. Consequently, the sensing rate of other sources are low.

In the second experiment, the value of $R^\#$ is increased from one to nine with an interval of two. In Figure 4.2, the max-min rate of $MILP$, $GND$ and $OUED$ grows with increasing $R^\#$. In particular, when $R^\#$ increases from one to nine, the max-min rate of $MILP$ increases by 52.75%; that is, from 40.42 to 61.74 kb/s. The max-min rate of $GND$ increases as much as 21.02 kb/s; i.e., from 40.42 to 61.44 kb/s. The max-min rate of $OUED$ increases from 40.20 to 61.55 kb/s. This is because more nodes are charged with increasing number of ACs. In addition, $UPPER$ and $LOWER$ stays at 61.45 and 22.47 kb/s, respectively. This is because $|V|$ and $\delta$ are fixed. Only the location of sources is changed. Another observation is that the

![Figure 4.1: Max-min rate with varying $|V|$.](image-url)
max-min rate cannot keep the linear increase with $R^\#$. Specifically, the max-min rate of MILP only increases by 2.38 kb/s from 59.36 kb/s when $R^\#$ increases from five to nine. As a comparison, the max-min rate of MILP increases as much as 18.94 kb/s when $R^\#$ increases from one to five. Additionally, when $R^\#$ increases from one to five, the max-min rate of GND and OUED increases to 18.21 and 17.87 kb/s, respectively. However, as $R^\#$ continues to increase to nine, the max-min rate of GND and OUED only increases by 2.81 and 3.48 kb/s, respectively. The reason is because link capacity restricts the increase of their max-min rate. In particular, as the total active time is one second; see constraint 4.4, after a given number of ACs, the max-min rate remains fixed. In terms of PATH, note that its max-min rate is smaller than LOWER. For example, when $R^\#$ is one, the max-min rate of PATH and LOWER is 11.17 and 24.69 kb/s, respectively. The reason is as follows. In order to obtain the max flow rate at the sink, PATH upgrades $R^\#$ nodes on the shortest path from each source to the sink; see Section 3.3.1. Thus, some sources may be left out. If these sources have a low energy harvesting rate, the max-min rate is low.

![Max-min rate with varying $R^\#$ values.](image)
4.5. Evaluation

The third experiment studies the effect of $|S|$ that increases from one to nine with an interval of two. Figure 4.3 shows that the max-min rate generated by each algorithm decreases as $|S|$ increases. Specifically, in the case of MILP, whereby $|S|$ increases from one to nine, the max-min rate reduces from 151.79 to 22.39 kb/s. The max-min rate of GND drops as much as 128.91 kb/s; i.e., from 151.30 to 22.39 kb/s. The max-min rate of OUED decreases from 142.28 to 22.35 kb/s, respectively. The value of UPPER and LOWER drops by 85.55% and 87.83% from 155.60 and 62.19 kb/s, respectively. The reason is as follows. First, as $|S|$ rises, there are more links used to transfer data. As a result, the total active time is shared by more links; see constraint 4.4. Thus, the link capacity of a link reduces; see constraint 4.3. Second, more sources generate data when $|S|$ rises. However, the energy harvesting rate of relay nodes is fixed such that a relay node transfers data from more sources. Consequently, the max-min rate reduces. Third, there are only three ACs. However, as $|S|$ increases, more nodes with low energy harvesting rate may be selected as sources. This also causes the max-min rate of PATH to drop from 124.57 to 0 kb/s. Note that the max-min rate of PATH becomes zero when there are seven or nine sources. The reason is as follows. PATH prefers to upgrade nodes on the shortest path from a source to the sink. As $|S|$ rises, the number of intermediate nodes on the shortest path from a source to the sink reduces. Thus, PATH may update all nodes on the shortest path from a source to the sink by only three ACs. Further, as the nodes on the path have ample energy, the LP solver prefers to allocate all active time to the links on this path in order to maximize the flow rate at the sink. Consequently, the sensing rate of the remaining sources becomes zeros.

The fourth experiment considers the impact of $\delta$; it takes on one of the following values: 3, 4, 5, 6 and 7. Referring to Figure 4.4, increasing $\delta$ has a positive impact on the max-min rate. In particular, as $\delta$ increases from three to seven, the max-min rate of MILP increases from 51.03 to 73.99 kb/s. The max-min rate of GND increases as much as 23.63 kb/s from 50.33 kb/s. The max-min rate of OUED also increases from 47.25 to 73.62 kb/s. The max-min rate of PATH also increases from 13.59 to
18.05 kb/s. This is due to the following reasons. First, as $\delta$ increases, the number of intermediate nodes on the shortest path from the sources and the sink decreases. Second, increasing $\delta$ means that a source has more paths to forward its data. For the same reasons, as $\delta$ increases from three to seven, $UPPER$ and $LOWER$ also increases by 30.67% and 42.16% from 57.94 and 19.07 kb/s, respectively. Another observation is that the max-min rate does not increase linearly with $\delta$. Specifically, $UPPER$ increases as much as 14.07 kb/s when $\delta$ increases from three to five. However, as $\delta$ increases to seven, $UPPER$ only increases by 5.14%. This is because the links surrounding the sink interfere with each other. Thus, the amount of data that the sink receives has an upper bound.

Next, the difference between the solution derived by MILP and the two heuristic algorithms; namely, $GND$, $OUED$ is analyzed. Referring to Figure 4.1 to 4.4, on average, the max-min rate of $GND$ attains 99.34% that of MILP. The reasons for the gap between $GND$ and MILP are as follows. First, after $GND$ deploys an AC in a given stage, a new routing is computed to maximize the min rate of all sources. However, under this new routing, the location of ACs deployed in previous stages
4.5. Evaluation

may no longer be optimal. Second, if there are several nodes that yield the highest increase in max-min rate, GND randomly upgrades one of these nodes.

The average gap between OUED and MILP is 1.72%. The reason is as follows. After OUED runs the LP solver, there may be several nodes that produce the same increase in max-min rate. Thus, OUED will randomly select a node from these nodes to upgrade. However, compared with GND, OUED does not ensure that the upgraded node produces the highest increase in max-min rate.

4.5.2 Large Networks

This subsection studies the performance of GND and OUED in large networks. In particular, these networks contain 200 nodes. The max-min rate of GND and OUED is recorded given varying number of ACs, number of sources and node degree. In addition, the running time of GND and OUED is also recorded. Note that each algorithm is run 60 times in each experiment. Note that PATH is also considered in these networks. However, the max-min rate of PATH is zero. Consequently, the following figures do not include results from PATH.
4.5. Evaluation

The first experiment explores the influence of $R^\#$. Its value increases from five to 25 with an interval of five. The value of $\delta$ and $|S|$ are three and 10 respectively. Referring to Figure 4.5, on average, the max-min rate of GND is smaller than that of OUED. In particular, on average, the max-min rate of OUED attains 99.76% that of UPPER. On the other hand, the max-min rate of GND achieves 99.22%. This indicates having more ACs increases the gap between MILP and GND. Another observation is that LOWER remains at 9.92 kb/s when $R^\#$ increases from five to 25. The max-min rate of GND and OUED stays at 20.30 and 20.41 kb/s, respectively. This indicates that five ACs are sufficient to maximize the minimum sensing rate. Referring to Figure 4.6, the running time of GND increases by 499.93%; i.e., from 81.08 to 486.42 kb/s. The running time of OUED also increases from 1.24 to 7.03 seconds. This is because both algorithms have $R^\#$ stages to deploy ACs; see Line 3 in Algorithm 4 and Line 2 in Algorithm 5.

![Figure 4.5: Max-min rate with varying $R^\#$ values in large networks.](image)

The next experiment studies the impact of $|S|$, which is increased from five to 25 with an interval of five. The value of $\delta$ and $R^\#$ is three and 10 respectively. Referring to Figure 4.7, the max-min rate of GND and OUED is 40.18 and 40.17
4.5. Evaluation

Figure 4.6: Running time with varying $R^#$ values in large networks.

kb/s, respectively, when $|S|$ is five. As expected, the max-min rate of $GND$ and $OUED$ reduces by as much as 31.95 and 31.94 kb/s, respectively, when $|S|$ increases to 25. In addition, on average, the max-min rate of $GND$ and $OUED$ reaches 99.98% and 99.97% that of $UPPER$. The reason is as follows. As per Figure 4.5, ten ACs are sufficient to increase the max-min rate to the upper bound. Thus, in this experiment, the link capacity restricts any further increase in max-min rate. Figure 4.8 shows the running time of $GND$ and $OUED$. When $|S|$ increases from five to 25, the running time of $GND$ increases by 46.05%, from 119.86 to 175.05 seconds. The running time of $OUED$ ranges from 3.21 to 3.57 seconds. The reasons are as follows. As $|S|$ increases, the number of decision variables in the LP of $GND$ and $OUED$ increases; see Proposition 6.

The last experiment studies the impact of $\delta$. Specifically, referring to Figure 4.9, when $\delta$ increases from three to seven, the max-min rate of both $GND$ and $OUED$ increases by 16.67%; from 21.24 to 24.78 kb/s. This indicates that increasing $\delta$ has a positive impact on the increase in max-min rate. In addition, the maximum gap between $GND$ and $UPPER$ is 0.22% when $\delta$ is three. This indicates that ten ACs
4.5. Evaluation

Figure 4.7: Max-min rate with varying $|S|$ values in large networks.

Figure 4.8: Running time with varying $|S|$ values in large networks.
are sufficient to upgrade nodes in the WSN. Another observation is that LOWER increases as much as 11.13 kb/s; i.e., from 7.65 to 18.78 kb/s when $\delta$ increases from three to seven. However, UPPER only increases by 3.49 kb/s from 21.29 kb/s. The reason is as follow. For LOWER, the energy harvesting rate of bottleneck nodes restricts the increase in max-min rate. As $\delta$ increases, there are more paths between a source and the sink. Thus, the source can select a path where its nodes have ample energy. However, in terms of UPPER, each node has ample energy. Consequently, only link capacity limits any further increase in max-min rate. Referring to Figure 4.10, the running time of GND increases from 121.87 to 2276.56 seconds when $\delta$ increases from three to seven. On the other hand, the running time of OUED also increases from 3.24 to 14.74 seconds. This is because increasing $\delta$ results in more links. Consequently, the number of decision variables increases; see Proposition 6.

![Graph](image)

**Figure 4.9:** Max-min rate with varying $\delta$ values in large networks.

### 4.6 Conclusion

This chapter studies fair rate allocation in energy harvesting WSNs. It proposes a novel problem called ACP-MM that upgrades a given number of sensor nodes in...
Figure 4.10: Running time with varying $\delta$ values in large networks.

order to maximize the minimum sensing rate of sources. The problem is modeled as a MILP. Two novel heuristics algorithms, namely $GND$ and $OUED$, are proposed to approximate the MILP solution. Evaluation results show that, on average, the max-min rate of $GND$ and $OUED$ attains 99.34% and 97.97% that of MILP. However, the running time of $OUED$ is significantly smaller than that of $GND$. Further, the experiments also show that increasing $|V|$ and $|S|$ have a negative impact on the max-min rate. By contrast, increasing $R^#$ and $\delta$ have a positive influence on the max-min rate.

Chapter 3 and 4 have studied AC deployment strategies in order to maximize throughput and maximize the minimum source rate, respectively. However, one AC only charges one sensor. Moreover, both solutions require the deployment of one or more ACs. Thus, in the next chapter, this thesis considers employing RF energy transmission to simultaneously charge multiple sensors. This allows a sensor node to transfer any excess energy to its neighbors with the aim to improve the max-min rate.
Energy Transfer and Max Min Rate

To date, see Section 2.1.2.1, past works have employed single hop energy transfer and investigated allocating one time slot for energy and information. They have also considered, see Section 2.1.2.2, delivering energy from a charger to nodes over multiple hops. Inspired by these previous works, this chapter combines time allocation and multi-hop energy transfer together. In particular, this chapter jointly considers energy sharing between nodes and data routing in a WSN comprising of nodes that harvest energy from both solar and RF. In addition, sensor nodes adopt a time switching architecture [68]. Given this setup, this chapter outlines the following so called RFES-MM problem: maximize the minimum rate of sources by determining the proportion of time used to transfer and receive energy as well as transfer and receive data. RFES-MM is modeled as a LP and solved using standard LP tools. In addition, this chapter outlines a centralized algorithm called CHMM. In particular, base on binary search, CHMM iteratively searches the maximum common sensing rate of all sources. In each iteration, each source has the same sensing rate and CHMM selects one fixed path for each source as a data transmission route. It then checks whether the set of paths and sensing rate meet all constraints. If yes, CHMM increases the sensing rate for the next iteration. Otherwise, it reduces the sensing rate. Apart from that, this chapter also studies the impact of transmit power, num-
ber of nodes, number of sources, conversion efficiency and number of sinks on the max-min rate.

The remainder of this chapter has the following structure. Section 5.1 describes the network model. Section 5.2 defines the problem formally. Section 5.3 presents the details of CHMM. Section 5.4 presents the evaluation results. Finally, Section 5.5 concludes the chapter.

5.1 Preliminaries

This section models a rechargeable WSN with \(|\mathcal{S}|\) sinks as a graph \(G(V \cup \mathcal{S}, E)\), where \(V\) and \(E\) denote the set of sensor nodes and directional links, respectively. All sensor nodes in \(V\) are uniformly distributed on a given field. There are \(|\mathcal{S}|\) source nodes, where \(\mathcal{S} \subseteq V\); these nodes generate data and transfer sensed data to the sink directly or via relay nodes. Each sensor node has data communication range \(r^I\). Let \(g_i\), in bits/s be the data generation rate of source node \(i\). Non source nodes act as relays and they are responsible for receiving and transferring data to the sink \(s\). Note, sink \(s\) is only able to receive data. Let \(d_{u,i}\) be the Euclidean distance between transmitter \(u\) and receiver \(i\). The set of communication neighbors of node \(i\) is recorded in the set \(\mathcal{I}_i^D = \{j \mid d_{i,j} \leq r^I, \forall j \in V \cup \mathcal{S}\}\). In other words, a node or a sink \(u \in \mathcal{I}_i^D\) is within the transmitting/receiving distance of node \(i\). Let \(f_{u,i}\) be the flow rate (bits/s) from node \(u\) to \(i\). The maximum link capacity is \(C\).

All nodes have two energy sources. First, each node is able to harvest energy from the environment; e.g., solar [16]. Critically, each node location has a different energy harvesting rate [16]. Let \(E_i\) be the energy harvesting rate of node \(i\). Second, a node also obtain energy from its neighbors via RF. The received power at a node is calculated by the Friis equation [116]. Specifically,

\[
P_{R_{u,i}} = P_t^T \frac{G_u G_i \lambda^2}{(4\pi d_{u,i})^2}
\]
where \( P_{u,i}^R \) is received power at node \( i \) from node \( u \), \( P_u^T \) is the transmit power of \( u \), \( G_u \) and \( G_i \) are the antenna gain of the transmitter \( u \) and receiver \( i \) respectively, and \( \lambda \) is the wavelength, respectively. For node \( i \) to harvest energy, the received power \( P_{u,i}^R \) must be at least larger than a given power sensitivity value; e.g., for the Powercast platform in [58], it is -11.5 dBm. Let \( r^E \) be the RF energy transmission range and \( I_E^i = \{ j \mid d_{i,j} \leq r^E, \forall j \in V \} \) are the neighbors from which node \( i \) can harvest energy from. Note, sinks cannot transfer RF energy to its neighbors. Further, received power is harvested with efficiency \( \eta_{u,i} \); this value is determined by the operating frequency, antenna, and matching network and rectenna [116]. Lastly, with regards to energy consumption, \( \rho, \sigma \) and \( \gamma \) denote, respectively, the power consumption rate due to receiving, sensing and transmitting (in Watt per bit).

Assume all nodes are synchronized and equipped with the time-switching architecture in [68]. Briefly, as a node cannot receive RF energy and information simultaneously, it thus must dedicate time to transmit/receive data/energy. To this end, the time slot of each node \( i \) is divided into the following sub-slots: (i) \( t_i^R \), the time used to receive data from neighbors, (ii) \( \tau_i^R \), the time used to receive energy from neighbors, (iii) \( t_i^T \), the time used to transfer data to neighbors, (iv) \( \tau_i^T \), the time used to transfer energy to neighbors. For node \( i \), its maximum RF harvested energy from all neighbors is \( \sum_{u \in I_E^i} \eta_{u,i} P_{u,i}^R \tau_i^R \). However, for a specific link \((u, i) \in E\), the maximum RF energy harvested from node \( u \) is determined by both \( \tau_i^R \) and \( \tau_u^T \). In particular, if \( \tau_i^R \) is larger than \( \tau_u^T \), the maximum RF energy that node \( i \) harvests from node \( u \) is equal to \( \eta_{u,i} P_{u,i}^R x_{u,i}^T \). Otherwise, node \( i \) can harvest at most \( \eta_{u,i} P_{u,i}^R x_{u,i}^R \) Joules from node \( u \). To capture these two scenarios, a decision variable called \( x_{u,i} \) is introduced. The amount of harvested RF energy by node \( i \) is \( \eta_{u,i} P_{u,i}^R x_{u,i}^T \). As mentioned, \( x_{u,i} \) must be smaller than or equal to \( \tau_i^R \) and \( \tau_u^T \).

Next is wireless interference. Define a matrix \( A \) where each row corresponds to a distinct data link. An entry in \( A \) is denoted by a binary variable \( a_{i,j}^n \). If \( a_{i,j}^n \) is one, the link \((i, j)\) is active in column \( n \). Each column of matrix \( A \) is an independent transmission set. That means the links in a set or a column are able
Table 5.1: Key Notations and Definitions in RFES-MM Problem

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{S}$</td>
<td>The set of sinks</td>
</tr>
<tr>
<td>$V$</td>
<td>The set of nodes</td>
</tr>
<tr>
<td>$E$</td>
<td>The set of links</td>
</tr>
<tr>
<td>$r^I$</td>
<td>Data communication range</td>
</tr>
<tr>
<td>$g_i$</td>
<td>Data generation rate at node $i$</td>
</tr>
<tr>
<td>$d_{u,i}$</td>
<td>Euclidean distance between transmitter $u$ and receiver $i$</td>
</tr>
<tr>
<td>$\mathcal{I}^C_i$</td>
<td>The set of communication neighbors of node $i$</td>
</tr>
<tr>
<td>$f_{ij}$</td>
<td>Number of bits that node $i$ transfers to node $j$</td>
</tr>
<tr>
<td>$C$</td>
<td>Link capacity</td>
</tr>
<tr>
<td>$E_i$</td>
<td>Energy harvesting rate of node $i$</td>
</tr>
<tr>
<td>$r^{E}$</td>
<td>RF energy transmission range</td>
</tr>
<tr>
<td>$\mathcal{I}^{E}_i$</td>
<td>The set of energy neighbors of node $i$</td>
</tr>
<tr>
<td>$P_u$</td>
<td>Transmit power at node $u$</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Energy consumption for receiving a bit</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Energy consumption for sensing a bit</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Energy consumption for transmitting a bit</td>
</tr>
<tr>
<td>$t^R_i$</td>
<td>Time used to receive data from neighbors</td>
</tr>
<tr>
<td>$\tau^R_i$</td>
<td>Time used to receive energy from neighbors</td>
</tr>
<tr>
<td>$t^T_i$</td>
<td>Time used to transfer data to neighbors</td>
</tr>
<tr>
<td>$\tau^T_i$</td>
<td>Time used to transfer energy to neighbors</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Conversion efficiency</td>
</tr>
<tr>
<td>$A$</td>
<td>Transmission sets</td>
</tr>
<tr>
<td>$S$</td>
<td>Superframe</td>
</tr>
</tbody>
</table>

to transmit simultaneously without causing interference at their respective receiver. For example, consider the topology A-B-C. As nodes cannot transmit and receive simultaneously, the matrix $A$ for this example contains two columns: $[1 \ 0]^T$ and $[0 \ 1]^T$. Note, the set of links in a column is determined by the protocol or physical interference model. The method of generating matrix $A$ is presented in Section 5.4. Next, there is a vector $S = [y_1, y_2, \ldots, y_N]$ representing the active time of each column in $A$. In particular, $y_n$ means the active time of column $n$ in matrix $A$. The total active time, i.e., $\sum_{y=1}^{N} = 1$, is one second. Table 5.1 lists common notations in this chapter.
5.2 Problem Definition

The aim is to maximize the minimum transmission rate of sources. To do this, for each node $i$, the value of $t_R^i$, $\tau_R^i$, $t_T^i$ and $\tau_T^i$ has to be determined. The problem is modeled as an LP with eight constraints.

The first constraint ensures that a decision variable called $x_{u,i}$ is equal to the minimum value between $\tau_T^u$ and $\tau_R^i$. Note, $x_{u,i}$ is used to calculate the amount of RF energy the node $i$ receives from node $u$; see constraint (5.2).

$$x_{u,i} = \text{MIN} (\tau_T^u, \tau_R^i), \forall i \in V, \forall u \in I$$

The second constraint ensures the harvested energy at node $i$ is larger than or equal to its consumed energy. Its harvested energy from the environment, e.g., solar, is $\mathcal{E}_i$. The RF energy harvested from node $u$ is determined by the minimum value of $\tau_T^u$ and $\tau_R^i$. Hence, considered the energy loss during transmission and power conversion, the amount of RF energy that node $i$ harvests from all neighbors is $\sum_{u \in I^E} \eta_{u,i} P_{u,i} x_{u,i}$. Thus, for all $i \in V$,

$$\mathcal{E}_i + \sum_{u \in I^E} \eta_{u,i} P_{u,i} x_{u,i} \geq \sum_{u \in I^P} \rho f_{u,i} + \sigma g_i + \sum_{u \in I^P} \gamma f_{i,u} + P^T \tau_T^i$$

The LHS represents the amount of energy that node $i$ harvests from both solar and its neighbors. The RHS corresponds to the energy consumption of node $i$. Note, if a node $i$ is a relay, then $\sigma g_i$ is zero.

The third constraint ensures each flow is conserved and thus ensures there is at least one connection from a source to the sink. In particular, for a node $i$, the sum of its received flow, denoted by $\sum_{u \in I^P} f_{u,i}$, and generated flow, denoted by $g_i$, is equal to its output flow, denoted by $\sum_{v \in I^P} f_{i,v}$. Formally, for each node $i \in V$,

$$\sum_{u \in I^P} f_{u,i} + g_i = \sum_{v \in I^P} f_{i,v}, \forall i \in V$$
The next constraint bounds the total demand on a link. In particular, the amount of data on a link \((u, i)\) is bounded by the total active time of this link as well as the maximum link capacity. In other words,

\[
f_{u,i} \leq \sum_{n=1}^{N} a_{n}^{u,i} C y_{n}, \forall i \in V, \forall u \in \mathcal{I}_{i}^{D} \tag{5.4}
\]

\[
f_{i,v} \leq \sum_{n=1}^{N} a_{n,i,v} C y_{n}, \forall i \in V, \forall v \in \mathcal{I}_{i}^{D} \tag{5.5}
\]

Note, constraint (5.4) represents the bound of an input flow of node \(i\). By contrast, constraint (5.5) means the bound of an output flow at node \(i\).

The following constraint bounds the total active time to be at most one second.

\[
\sum_{n=1}^{N} y_{n} = 1, \forall i \in V, \forall u \in \mathcal{I}_{i}^{D} \tag{5.6}
\]

Next, the time dedicated to receiving and transmitting information at node \(i\) must be larger than the active time of its any adjacent link. As an example, consider two nodes \(u\) and \(v\) that are transferring data to node \(i\). Assume that the active time of link \((u, i)\) is 0.4 second. To receive data, the value of \(t_{u}^{T}\) and \(t_{i}^{R}\) is at least 0.4 second. Assume the active time of link \((v, i)\) is 0.6 second. To receive data, the value of \(t_{v}^{T}\) and \(t_{i}^{R}\) is at least 0.6 second. Thus, \(t_{u}^{T}\) must be larger than 0.4 second. Similarly, the value of \(t_{v}^{T}\) and \(t_{i}^{R}\) has to be larger than 0.6 second. To model the foregone scenario, we thus have, Formally,

\[
\sum_{n=1}^{N} a_{n,u,i} y_{n} \leq MIN(t_{u}^{T}, t_{i}^{R}), \forall i \in V, \forall u \in \mathcal{I}_{i}^{D} \tag{5.7}
\]

For each node \(i\), its total time dedicated to receive energy from chargers plus transmission and reception time must be no more than one. In other words,

\[
\tau_{i}^{T} + \tau_{i}^{R} + t_{i}^{T} + t_{i}^{R} \leq 1, \forall i \in V \tag{5.8}
\]
Lastly, the problem is modeled as the following formulated LP.

\[
\begin{align*}
\text{MAX} & \quad \min_{i \in S} \{g_i\} \\
\text{subject to} & \quad (5.1), (5.2), (5.3), (5.4), (5.5), (5.6), (5.7), (5.8)
\end{align*}
\]

Next, the number of decision variables and constraints are analyzed. Both of which have an influence on the LP’s computation time.

**Proposition 13.** In the formulated LP, there are \(2|E| + |S| + 4|V| + N\) decision variables.

*Proof.* The decision variables are \(x_{u,i}, f_{u,i}, g_i, y_n, t^R_i, \tau^R_i, t^T_i\) and \(\tau^T_i\). There are at most \(|E|\) decision variables of type \(x_{u,i}\) and \(f_{u,i}\). The number of variable \(g_i\) corresponds to \(|S|\) sources. Each column of matrix \(A\) has a decision variable; i.e., \(y_n\). Lastly, each node has the decision variables \(t^R_i, \tau^R_i, t^T_i\) and \(\tau^T_i\). In total, there are \(2|E| + |S| + 4|V| + N\) decision variables. □

**Proposition 14.** In the formulated LP, there are at most \(5|V| + 3|E| + 1\) constraints.

*Proof.* First, all \(|V|\) nodes have energy and flow conservation constraints. Second, each decision variable \(x_{u,i}\) has constraints (5.1). Hence, There are at most \(2|E|\) constraints for \(x_{u,i}\). Additionally, in the set \(E\), each link has one link capacity constraint; see constraints (5.4) and (5.5). Next, each node has a constraint to bound its data receiving and transferring time, respectively; see constraints (5.7). There is constraint (5.8) for each node. Lastly, there is one constraint (5.6). Consequently, the number of constraints is at most \(5|V| + 3|E| + 1\). □

### 5.3 A Heuristic Solution

This section outlines a centralized heuristic max-min rate (CHMM) algorithm. *CHMM* employs binary search to find the maximum common sensing rate that
meets constraints (5.1)-(5.8). In each iteration, \( CHMM \) calculates a common sensing rate and checks whether it is feasible. It includes two steps: (i) \textit{path select}, where \( CHMM \) selects one data transmission path for each source to the sink and generates the matrix \( A \) containing only links used by these paths, and (ii) \textit{constraint test}, where \( CHMM \) checks whether the current sensing rate is feasible using a revised LP.

The details of \( CHMM \) are as follows. Let \( \hat{g} \) represent the common sensing rate of all sources. Let \( LB \) and \( UB \) be respectively the lower and upper sensing rate bound. At the beginning of each iteration, \( CHMM \) updates \( \hat{g} \) to the average value of \( LB \) and \( UB \). Then it enters the \textit{path select} step. For each source node, \( CHMM \) employs Yen’s algorithm \[145\] to generate three paths to the sink. Assume that each node has already received RF energy from its all neighbors. \( CHMM \) then finds the node with the minimum energy on each path. The source node then uses the path with the highest minimum energy to transfer data to the sink. Further, for the nodes on the selected path, \( CHMM \) updates their energy by subtracting the energy used to forward \( \hat{g} \). Then, \( CHMM \) generates a matrix \( A \), using the algorithm in Section 3.4, that includes only links of these paths.

The next step is \textit{constraint test}. First, \( CHMM \) revises the formulated LP in Section 5.2. Specifically, given \( \hat{g} \), \( CHMM \) removes its objective and then uses a fixed sensing rate \( \hat{g} \) instead the decision variable \( g_i \), where \( CHMM \) modifies constraint (5.2) and (5.3) as follows.

\[
\sum_{u \in I_i^D} \rho f_{u,i} + \sum_{u \in I_i^D} \gamma f_{i,u} + P_i^T \tau_i^T - \sum_{u \in I_i^R} \eta_{u,i} P_{u,i} x_{u,i} \leq E_i - \sigma \hat{g} \tag{5.9}
\]

\[
\sum_{v \in I_i^D} f_{i,v} - \sum_{u \in I_i^D} f_{u,i} = \hat{g}, \forall i \in V \tag{5.10}
\]
Consequently, the revised LP is as follows.

\[
\begin{align*}
\text{MAX} & \quad MIN\{g_i\}_{i \in S} \\
\text{subject to} & \quad (5.1), (5.9), (5.10), (5.4), (5.5), (5.6), (5.7), (5.8)
\end{align*}
\]

Second, \textit{CHMM} solves the revised LP to check whether \( \hat{g} \) is feasible. If \( \hat{g} \) is feasible, \textit{CHMM} increases the value of \( LB \) to \( \hat{g} \). Otherwise, \textit{CHMM} decreases the value of \( UB \) to \( \hat{g} \). When the difference between \( UB \) and \( LB \) is smaller than 0.001, \textit{CHMM} stops.

Algorithm 6 is used to explain \textit{CHMM} in more details. Line 2 sets \( \hat{g} \) to the average value of \( LB \) and \( UB \). Next, \textit{CHMM} calls the function \textit{UpdateEnergy()} to calculate the energy of each node when its neighbors transfer all energy to it via RF; see Line 3. Note that \textit{UpdateEnergy()} returns a set denoted by \( E' \) containing the current energy of all nodes. For each source node \( i \), \textit{CHMM} calls the function \textit{Yen()} to obtain three shortest paths from the source node \( i \) to the sink. It uses \( p \) to store the three paths. In addition, it also uses \( w \) to record the minimum energy of nodes on each path; see Line 5. Next, \textit{CHMM} calls the function \textit{SelectPath()} in order to select the path with the highest minimum energy; see Line 6. It returns the selected path \( p^*_i \) for source node \( i \). In Line 7, \textit{CHMM} calls the function \textit{SubEnergy()} to subtract the energy consumption by each node on the path \( p^*_i \) with a data transmission rate of \( \hat{g} \). Next, \textit{CHMM} adds the path \( p^*_i \) into the set \( P \); see Line 8. When each source node has a path, \textit{CHMM} then calls the function \textit{UpdateTopology()} to generate a topology that only includes links on the selected paths; see Line 10. Let the new topology containing only these links be \( G(V', E') \), where \( V' \) and \( E' \) denotes the set of nodes and the links on the selected paths, respectively. In order to check whether \( \hat{g} \) meets all constraints, Line 11 calls the function \textit{ConstraintsOK()}. In particular, this function calls a LP solver to solve the revised LP. If the revised LP is feasible, this causes \textit{CHMM} to increase \( LB \) to \( \hat{g} \); see Lines 12. Otherwise, if the revised LP is not feasible, \textit{CHMM} reduces the \( UB \) to \( \hat{g} \); see Lines 14.
Algorithm 6: CHMM Algorithm

Input: \( LB, UB, G(V, E), P^T, \eta, \mathcal{E}, \mathcal{S}, A \)

Output: \( \hat{g} \)

1. while \( UB - LB \geq 0.001 \) do
   2. \( \hat{g} = (UB - LB)/2; \)
   3. \( \mathcal{E}' = UpdateEnergy(\mathcal{E}, P^T, \eta, G(V, E)); \)
   4. foreach \( i \) in \( \mathcal{S} \) do
      5. \([p, w] = Yen(i); \)
      6. \( p^*_i = SelectPath(p, w, \mathcal{E}'); \)
      7. \( \mathcal{E}' = SubEnergy(p^*_i, w, \mathcal{E}'); \)
      8. \( P = AddPath(p^*_i); \)
   9. end
   10. \( G(V', E') = UpdateTopology(P); \)
   11. if \( ConstraintsOK(\hat{g}, A, G(V', E')) \) then
      12. \( LB = \hat{g}; \)
   13. else
      14. \( UB = \hat{g}; \)
   15. end
16. end

Proposition 15. The time complexity of CHMM is \( \mathcal{O}(|\mathcal{S}||V| \log(|UB - LB|)) \).

Proof. In Line 3, there are at most \(|V| \) that updates their energy. Line 4 runs \(|\mathcal{S}| \) times. As per [145], the time complexity of Yen’s algorithm in Line 5 is \( \mathcal{O}(3|V|^3) \). Next, as there are three paths and at most \(|V| \) nodes on each path, Line 6 takes \( \mathcal{O}(3|V|^3) \) time to select the path with the highest minimum energy. Further, in Line 7, the function \( SubEnergy() \) takes at most \( \mathcal{O}(|V|) \) to update the energy of nodes on the path. Thus, the time complexity of Line 4-9 is \( \mathcal{O}(|\mathcal{S}||V|^3) \). Next, in Line 10, the worst case is that each pair of nodes has a link such that the function \( UpdateTopology() \) takes at most \( \mathcal{O}(|V|^2) \). As there are \( 5|V| + 3|E| + 1 \) constraints to check, the function \( ConstraintsOK() \) has a time complexity of \( \mathcal{O}(5|V| + 3|E| + 1) \). The most expensive part is \( \mathcal{O}(|\mathcal{S}||V|^3) \). In addition, the time complexity of binary search in Line 1 is \( \mathcal{O}\log(|UB - LB|) \). Thus, the time complexity of \( CHMM \) is \( \mathcal{O}(|\mathcal{S}||V|^3 \log(|UB - LB|)) \).
Table 5.2: Evaluation parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$</td>
<td>500 kbps</td>
</tr>
<tr>
<td>$\rho$</td>
<td>93.5 nJ/b</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>93.5 nJ/b</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>150 nJ/b</td>
</tr>
<tr>
<td>$E_i$</td>
<td>0-75 mW</td>
</tr>
<tr>
<td>$G_t$</td>
<td>6.1 dBi</td>
</tr>
<tr>
<td>$G_r$</td>
<td>6.1 dBi</td>
</tr>
</tbody>
</table>

5.4 Evaluation

All simulation is conducted in Matlab [138]. Sensor nodes are randomly scattered on a $5 \times 5$ m$^2$ sensing area. Each node knows the distance between itself and nodes. Without loss of generality, the simulation uses the parameters in [108] where sensor nodes are equipped with a +6.1 dBi antenna and a TI CC1101 transceiver [5] that has a theoretical capacity of 500 kbps. The receiver sensitivity required to decode information is assumed to be -31.5 dBm. Further, to receive RF energy, each node also has a P2110 RF Powercast receiver [146] that operates at a frequency of 915 MHz, has a power sensitivity of -11.5 dBm and conversion efficiency of 55%. A sensor node is also equipped with an EnOcean ECS310 solar cell with a maximum recharging rate of 150 mW in daytime and a minimum recharging rate of zero at night. The energy harvesting rate is a sinusoidal function based on the real solar irradiance data retrieved from Southwest Solar Research Park, Phoenix, Arizona, USA [144] obtained on the 16-th April 2013; on average, the energy harvesting rate of a node ranges from 0 to 75 mW. With regard to energy consumption, as per [5], the value of $\rho$ and $\gamma$ is 93.6 nJ/b, and 103.2 nJ/b, respectively, where the transmit power is +0 dBm [5]. The value of $\sigma$ is set to 150 nJ/b [132]. Table 5.2 summarizes the parameters used in the evaluation.

Matrix $A$ is generated as follows. Let $Z$ denote a transmission set or vector with dimension $|E| \times 1$. Let $B$ be a set that records all links in $E$. A link is randomly selected from the set $B$, say $l$, and added into vector $Z$. All links that conflict, as per
the protocol interference model [133], with $l$ are removed. These steps are repeated to select the next random link until there are no remaining links. Second, vector $Z$ is added into matrix $A$. The links recorded in vector $Z$ are removed from the set $B$. This process terminates when each row in $A$ has at least one entry with a value of one.

5.4.1 Results

This section studies the influence of five parameters: number of nodes $|V|$, number of sources $|S|$, transmit power $P_T$, conversion efficiency $\eta$ and number of sinks $|S|$. In each experiment, one parameter varies whilst the others are fixed. The results are an average of 100 runs with randomly selected sources in each experiment. As a comparison, let $g^{RF}$ be the max-min rate when each sensor node has ability to harvest energy from its neighbors and solar. On the other hand, $g^{NoRF}$ refers to the max-min rate when each node only harvests energy from solar. Let $g^{CHMM}$ be the max-min rate obtained by CHMM. In addition, to aid presentation, let $R_E$ and $R_E$ represent the average RF energy receiving time of all nodes obtained by the formulated LP and CHMM, respectively. Let $T_E$ and $T_E$ be the average RF energy transmission time of all nodes calculated by the formulated LP and CHMM, respectively. Next, the average data receiving time obtained by the formulated LP and CHMM is denoted by $R_D$ and $R_D$, respectively. On the other hand, the average data transmission time obtained by the formulated LP and CHMM is denoted by $T_D$ and $T_D$, respectively.

The first experiment studies the impact of $|S|$. In this respect, $|S|$ increases from one to nine with an interval of two. The value of $|S|$, $|V|$, $P_T$ and $\eta$ is set to one, 30, +12 dBm and 55% respectively. As per Figure 5.1, $g^{RF}$ reduces from 151.50 to 25.72 kb/s. The value of $g^{NoRF}$ drops from 143.53 to 20.02 kb/s. The value of $g^{CHMM}$ decreases from 147.63 to 25.58 kb/s. The reason is because more sources generate data but the solar energy harvesting rate of relay nodes is fixed. Further,
as $|S|$ increases, the number of relay nodes drops when $|V|$ is fixed. Consequently, fewer nodes transfer RF energy to their neighbors in high $|S|$ scenarios. Another reason is as follows. In high $S$ scenario, more links are used to transfer data. Hence, the total active time is allocated with more links. For example, if a source directly connects to the sink, all active time can be allocated to the link of the sink and the sink. However, if nine sources directly connect to the sink, the active time has to be allocated to nine links. As a result, the sensing rate at each source drops. In addition, note that $g^{RF}$ has better performance than $g^{NoRF}$ in high $|S|$ scenarios. Specifically, $g^{NoRF}$ achieves 94.73% of $g^{RF}$ when there is only one source. In scenarios with nine sources, $g^{NoRF}$ only attains 77.85% of $g^{RF}$. With increasing number of sources, there will be more sources that have a low energy harvesting rate, which in turn have a negative impact on $g^{NoRF}$. If these sources can harvest energy from their neighbors, the max-min rate rises; see Equ 5.2. Another observation is that the value of $g^{RF} - g^{CHMM}$ reduces from 3.87 to 0.14 kb/s when $|S|$ increases from one to nine. This is because $CHMM$ includes more data transmission paths as $|S|$ increases. The reason of the gap between $g^{RF}$ and $g^{CHMM}$ is because $CHMM$ does not consider the sensing rate of other sources when it selects a path for a source. For example, one source $s_1$ can select path $a$ or path $b$ to transfer data at a data rate. Another source $s_2$ only can select path $a$ to transfer data at this data rate. Thus, if $s_1$ selects path $a$, $s_2$ cannot ensure the data rate. However, if $s_1$ selects path $b$, the max-min rate of $s_1$ and $s_2$ can reach the data rate. Additionally, the results show that $g^{CHMM}$ may be smaller than $g^{NoRF}$ in one source scenario. The reason is as follows. $CHMM$ only has one data transmission path when $|S|$ is one. Thus, the max-min rate is restricted by the node with the minimum energy on the path. In contrast, $g^{NoRF}$ has multiple paths to transfer data.

In Figure 5.2, $T_E$ is approximate 2.32 times longer than $R_E$. In addition, $R_E$ only attains 27.03% of $T_E$. The reason is as follows. To increase the max-min rate of all sources, bottleneck nodes receives energy from their neighbors. Note that the number of bottleneck nodes is significantly smaller than that of their neighbors.
Thus, the number of nodes receiving energy is smaller than that of nodes transferring energy. Additionally, the value of $R_E$ and $T_E$ increased by 86.77% and 29.01% from 0.133 and 0.358 seconds, respectively. The reason is as follows. When there is one source, its total flow rate increase is $g^{RF} - g^{NoRF}$; i.e., 7.98 kb/s. However, when there are nine sources, it is at least $|S|(g^{RF} - g^{NoRF})$; i.e., 51.30 kb/s. Thus, more generated data indicates there is more energy transfer among nodes to increase the max-min flow when there are many sources. Consequently, both $R_E$ and $T_E$ increase with $|S|$. Another observation is that, on average, $R_E + T_E$ is 0.614 seconds. However, $R_E + T_E$ is 0.942 seconds. This is because CHMM only consider one path for each source to transfer data. Consequently, in one time slot, the nodes without data transmission can dedicate all transmission and reception time to energy transfer.

Referring to Figure 5.3, $T_D$ increases from 0.034 to 0.048 seconds. This is because the increase in $|S|$ results in more data, meaning nodes require more time to transfer data. It is also the reason why $T_D$ increases from 0.016 to 0.026 seconds. $R_D$ reduces from 0.474 to 0.241 seconds. This is because $R_E$ and $T_E$ increase when $|S|$ rises; see
Equ 5.8. In particular, as per Equ 5.3, the received flow of a node cannot be larger than the transferred flow of the node. However, in Figure 5.3, \( R_D \) is approximately 8.50 times higher than \( T_D \). This indicates each node \( i \) does not use all \( t_i^R \) to receive data; see Equ 5.7. Additionally, \( R_D \) reduces 0.006 to 0.004 seconds. There are two reasons. First, as per Figure 5.1, the sensing rate of each source reduces with the increase of \(|S|\). Moreover, in high \(|S|\) scenario, more sources directly connect to the sink such that their time used to receive data is zero. As a result, \( R_D \) reduces when \(|S|\) rises. This is also the reason why \( T_D \) is average 4.34 times longer than \( R_D \).

Referring to Figure 5.4, the running time of the formulated LP increases from 1.79 to 4.90 seconds. This is because the number of decision variable \( g_i \) increases as \(|S|\) rises; see Proposition 13. The running time of CHMM increases by 209.68%; i.e., from 1.55 to 4.80 seconds. This is because CHMM employs Yen’s algorithm to generate paths and more sources means more paths; see Line 5 in Algorithm 6.

The second experiment studies the impact of transmit power. Similar to the last experiment, there is only one sink. \(|V|\) and \( \eta \) is 30 and 55%, respectively. Also, \(|S|\) has a value of five. Referring to Figure 5.5, \( g^{RF} \) increases from 32.89 to 49.99 kb/s as
5.4. Evaluation

Figure 5.3: The value of $R_D$, $T_D$, $R_D$ and $T_D$ with varying $|\mathcal{S}|$.

Figure 5.4: Running time with varying $|\mathcal{S}|$. 

7. Running time (Seconds)
transmit power increases from +5 dBm to +25 dBm with a step size of five. \( g^{CHMM} \) increases by 25.93%; i.e., from 32.24 to 48.55 kb/s. This is because a higher transmit power allows a receiver to have a higher RF energy harvesting rate; see Equ. 5.2. Another reason is that a higher transmission power increases energy transmission range. In particular, when transmit power increases from +5 to +25 dBm, energy transmission range increases from 0.71 to 7.10 meters. Consequently, a node has more neighbors. However, as the topology and energy harvesting rate of each node are fixed, \( g^{NoRF} \) approximately stays at 29.86 kb/s when transmit power increases. Another observation is that the value of \( g^{RF} - g^{CHMM} \) increases by 121.54%; from 0.65 to 1.44 kb/s. The reason is as follows. As per Figure 5.5, \( g^{CHMM} \) increases with the transmit power. However, \( CHMM \) only selects one path for each source. Consequently, flow rate on these paths may be restricted by the link capacity.

In Figure 5.6, \( T_E \) increases from 0.092 to 0.599 seconds when transmit power increases from +5 dBm to +15 dBm. However, as transmit power continue to increase to +25 dBm, \( T_E \) drops by as much as 77.13%. The reasons are as follows. First, the solar energy harvesting rate of each node is fixed. If the transmit power
rises, the time used to transmit RF energy at a node drops. Further, a node with a higher transmit power spends less RF transmitting time on meeting the energy requirement of a bottleneck node. On the other hand, $R_E$ increases by 284.24%; i.e., from 0.184 to 0.707 seconds when the transmit power increases from $+5$ dBm to $+25$ dBm. This is because more nodes prefer to receive energy in order to increase the max-min rate when one of their neighbors has a high RF transmit power. Based on the same reasons, $R_E$ and $T_E$ has the similar tendency with the $R_E$ and $T_E$ respectively. In particular, when transmit power increases from $+5$ to $+15$ dBm, $T_E$ increases from 0.201 to 0.654 seconds. It then drops by 80.73%. On the other hand, $R_E$ increases by 478.77%; i.e., from 0.146 to 0.845 seconds.

In Figure 5.7, $T_D$ increases from 0.032 to 0.058 seconds when the transmit power increases from $+5$ dBm to $+25$ dBm. This indicates that nodes transfer more data when they receive more energy from their neighbors. Another observation is that $R_D$ reduces by 95.09%; i.e., from 0.692 to 0.034 seconds, when the transmit power increases from $+5$ dBm to $+25$ dBm. Note that, on average, $R_D$ is larger than $T_D$. This indicates that node $i$ is allocated a high $t_i^R$ but node $i$ only use a part of it to

![Figure 5.6: The value of $R_E$, $T_E$, $R_E$ and $T_E$ with varying $P_T$.](image)
receive data. Referring to Figure 5.7, as some sources directly connect to the sink, $T_D$ is at least 2.7 times longer than $R_D$. In addition, Figure 5.6 and Figure 5.7 also indicates that nodes allocate more time for energy exchanging with neighbors when transmit power rises. For example, $T_D + R_D$ reduces from 0.725 to 0.092 seconds, $T_D + R_D$ reduces from 0.056 to 0.029 seconds. On the other hand, $T_E + R_E$ increases from 0.275 to 0.844 seconds. $T_E + R_E$ increases from 0.347 to 0.971 seconds.

![Figure 5.7: The value of $R_D$, $T_D$, $R_D$ and $T_D$ with varying $P_T^u$.](image)

In Figure 5.8, the running time of CHMM increases from 2.70 to 5.13 seconds. On the other hand, the running time of the formulated LP also increases as much as 2.94 seconds from 6.03 seconds. This is because the increase of transmit power increases the number of decision variable $x_{i,j}$. In other words, each node receives energy from more neighbors.

The third experiment studies the impact of $\eta$. The value of $|S|$, $|V|$, $|S|$ and $P_i^T$ is set to one, 30, five and +12 dBm, respectively. Figure 5.9 shows that $g^{RF}$ increases from 45.44 kb/s to 52.81 kb/s when $\eta$ increases from 20% to 100%. The value of $g^{CHMM}$ increases by 15.02%; i.e., from 43.39 to 49.91 kb/s. This is because a higher conversion efficiency increases the amount of RF harvested energy at a
5.4. Evaluation

Transmit power (dBm):
- 5
- 10
- 15
- 20
- 25

Running time (Seconds):
- 2.5
- 3
- 3.5
- 4
- 4.5
- 5
- 5.5
- 6
- 6.5

Running time of CHMM
Running time of LP

Figure 5.8: Running time with varying $P_u^T$.

node; see Equ 5.2. In addition, since the topology and solar energy harvesting rate of each node are fixed, $g^{NoRF}$ approximately stays at 44.39 kb/s when $\eta$ increases from 20% to 100%.

In Figure 5.10, $R_E$, $T_E$, $R_E$ and $T_E$ stays at 0.093, 0.374, 0.211 and 0.755, respectively. This is because, for each node $i$, its energy neighbors $I_i^E$ is fixed when $\eta$ rises. Additionally, as expected, the time used to transfer energy is higher than the time used to receive energy in both CHMM and the formulated LP. In particular, the ratio of $T_E$ and $R_E$ is 4.02. The ratio of $T_E$ and $R_E$ is 3.58.

Referring to Figure 5.11, since $\eta$ has no impact on the communication neighbors of each node, $R_D$, $T_D$, $R_D$ and $T_D$ stays at 0.019, 0.032, 0.004 and 0.029 seconds, respectively. In addition, note that $R_D$ attains 59.38% of $T_D$ and $R_D$ achieves 13.79% of $T_D$. This also indicates that some source node directly connect to the sink.

Referring to Figure 5.12, on average, the running time of the formulated LP stays at 2.62 seconds. This is because $\eta$ has no impact on the number of decision variables; see Proposition 13. Additionally, the running time of CHMM stays at 2.29
5.4. Evaluation

Figure 5.9: Max-min rate with varying $\eta$.

Figure 5.10: The value of $R_E$, $T_E$, $R_{\text{CHMM}}$ and $T_{\text{CHMM}}$ with varying $\eta$. 
5.4. Evaluation

Conversion efficiency

\begin{align*}
\text{Time used to data (Seconds)}
\end{align*}

\begin{align*}
10^{-3} & \quad 10^{-2} & \quad 10^{-1}
\end{align*}

\begin{align*}
\sigma & \quad \tau & \quad \sigma & \quad \tau
\end{align*}

Figure 5.11: The value of $R_D$, $T_D$, $R_D$, and $T_D$ with varying $\eta$.

seconds which is smaller than that of the formulated LP. This is because CHMM considers fewer data transmission links than the formulated LP.

The fourth experiment studies the following values of $|V|$: 10, 20, 30, 40 and 50. The parameters $|\mathcal{G}|$, $|\mathcal{S}|$, $P_T^i$ and $\eta$ is set to one, five, +12 dBm and 55%, respectively. Referring to Figure 5.13, $g^{RF}$ increases as much as 25.40 kb/s; i.e., from 28.78 to 54.18 kb/s. On the other hand, $g^{NoRF}$ recorded an increase of 55.97%; i.e., from 28.16 kb/s to 43.92 kb/s. This is because the number of paths from a source to the sink increases. For a source having ample energy, more paths to the sink means it can transfer more data to the sink. Another reason is that, in the fixed sensing area, the increase of $|V|$ results that a node has more neighbors which transfer energy to it. This is also the reason why $g^{CHMM}$ increases from 28.77 to 53.52 kb/s when $|V|$ increases from 10 to 50. Another observation is that the value of $g^{RF} - g^{NoRF}$ increases from 0.62 kb/s to 10.25 kb/s when $|V|$ increases from 10 to 50. This shows that a higher node density has positive impact on further increasing the max-min rate of all sources when each node can harvest from its neighbors. The reasons are as follows. First, since $|V|$ increases in a fixed sensing area, a bottleneck
node can harvest RF energy from more nodes. Consequently, a node with a high solar harvesting rate can charge more nodes. Further, increasing $|V|$ also reduces the average distance between any two nodes. This thus results in higher received RF power, meaning nodes can harvest more energy; as per [5], a higher received power results in a higher conversion efficiency. Note that the value of $g^{RF} - g^{CHMM}$ increases from 0.01 to 0.65 kb/s. The reason is as follows. When $|V|$ increases, each node has more paths to the sink. However, $CHMM$ only selects one path for each source.

Figure 5.14 shows that $R_E$ reduces from 0.489 to 0.065 seconds when $|V|$ increases from 10 to 50, and $T_E$ increases by 108.23%; i.e., from 0.194 to 0.403 seconds. The reason is because there are more nodes that transfer their energy to bottleneck nodes when $|V|$ increases. Based on the same reason, $R_E$ and $T_E$ has the similar tendency with $R_E$ and $T_E$, respectively. Specifically, $R_E$ reduces from 0.345 to 0.199 seconds. $T_E$ increases from 0.234 to 0.771 seconds. In addition, note that $T_E + R_E$ and $T_E + R_E$ is 0.579 and 0.682 seconds, respectively, when $|V|$ is ten. However, as $|V|$ increases to 90, the former becomes approximate two times higher than the
latter. The reason is because the increase of $|V|$ has limited impact on the topology $G(V', E')$; see Algorithm 6. Hence, as $|V|$ increases, more nodes allocate their all time for transferring energy in order to increase $g^{CHMM}$.

Referring to Figure 5.15, $T_D$ reduces from 0.079 to 0.018 seconds. $T_D$ drops by 85.71%; i.e., from 0.140 to 0.020 seconds. This indicates that the increased nodes prefer to transfer energy when $|V|$ increases. In addition, $R_D$ reduces from 0.033 to 0.003 seconds, which corresponds the decrease of $T_D$. With regard to $R_D$, note that it has no significant tendency. When $|V|$ increases from 10 to 40, $T_D$ at most attains 33.29% of $R_D$. However, $R_D$ then sharply drops from 0.359 to 0.009 seconds. This indicates that the LP solver allocates more time of receiving data to nodes than the active time of these nodes’ adjacent links when $|V|$ increases from 10 to 40.

In Figure 5.16, the running time of CHMM and the formulated LP is 0.300 and 0.301 seconds respectively when $|V|$ is ten. As $|V|$ increases to 50, the running time of CHMM and the formulated LP increase as much as 11.42 and 12.95 seconds, respectively. This is because the increase of $|V|$ results in the increase of decision variables $f_{u, i}$, $y_n$, $t_i^R$, $T^R_i$, $T^T_i$ and $T^T_i$. In addition, as per Proposition 15, CHMM runs
Figure 5.14: The value of $R_E$, $T_E$, $R_E$ and $T_E$ with varying $|V|$.

Figure 5.15: The value of $R_D$, $T_D$, $R_D$ and $T_D$ with varying $|V|$.
a longer time when $|V|$ rises.

![Figure 5.16: Running time with varying $|V|$.

The last experiment studies the impact of $|S|$ which increases from one to nine. The value of $|V|$, $|S|$, $\eta$ and $PT$ is set to 30, five, 55% and +12 dBm, respectively. Additionally, note that all $|S|$ sinks are randomly selected from $|V|$ nodes. Further, with regards to CHMM, it selects one path for each source. In particular, for a source, CHMM employs Yen’s algorithm to generate three paths from it to each sink. The source then selects only one path via the function $SelectPath()$ to transfer its data; see Line 6 in Algorithm 6.

Referring to Figure 5.17, $g^{RF}$ increases as much as 19.06 kb/s; i.e., from 52.31 to 71.37 kb/s when $|S|$ increases from one to seven. The value of $g^{NoRF}$ increases from 47.12 to 66.12 kb/s. The value of $g^{CHMM}$ also increases by 23.37% from 49.26 to 60.77 kb/s. The reason is as follows. When there is one sink, all sources transfer data to the sink such that there is a heavy interference near the sink. In nine sinks scenario, sources can transfer their data to different sinks in order to reduce interference. In addition, more sinks means a source may be next to a sink. Thus, the source can transfer data directly to the sink. Another observation is that
5.4. Evaluation

g_{RF}, g_{NoRF} and g_{CHMM} drops to 2.71, 1.51 and 5.04 kb/s, respectively, when |S| increases from seven to nine. This is because more nodes are selected as sinks. In this respect, they cannot transfer energy to their neighbors. Note that g_{CHMM} is higher than g_{NoRF} when |S| increases from one to three. However, if |S| continues to increase, g_{CHMM} is smaller than g_{NoRF}. For example, when |S| is nine, the value of g_{CHMM} − g_{NoRF} is as much as 8.88 kb/s. This is because CHMM only selects one data transmission path for each source.

Figure 5.17: Max-min rate with varying S.

Figure 5.18 shows that T_E, R_E, T_E and R_E reduce with increasing |S|. Specifically, T_E reduces from 0.359 to 0.207 seconds. T_E drops from 0.653 to 0.392 seconds. This also indicates that more sinks means fewer nodes can transfer energy to their neighbors. Based on the same reason, R_E and R_E drop by 20.14% and 15.04%, from 0.144 and 0.226 seconds, respectively.

In Figure 5.19, T_D and T_D stay at 0.040 and 0.052 seconds, respectively. This is because higher |S| values have both positive and negative impact on the data transmission time. In particular, more sinks means more data is collected; see Figure 5.17. Further, if a source is next to the sink, it only transmits data to the sink such
that it only has data transmission time. However, as more nodes are selected as sinks, the increasing $|\mathcal{S}|$ has a negative impact on the average data transmission time of all nodes.

Referring to Figure 5.20, the running time of $CHMM$ increases as much as 6.1 seconds; i.e., from 2.33 to 8.43 seconds. This is because $CHMM$ calls Yen’s algorithm to obtain three paths between each pair of source and sink. However, the running time of the formulated LP reduces from 3.04 to 0.97 seconds. This is because the number of decision variables $f_{u,i}$, $x_{u,i}$ and $y_n$ reduce when $|\mathcal{S}|$ nodes are selected as sinks.

5.5 Conclusion

This chapter has studied a novel problem called RFES-MM. Its aim is to determine the time used for energy sharing and data transmission at each node in order to maximize the minimum sensing rate. It models the problem as a LP. As a comparison, a heuristic algorithm called $CHMM$ is proposed. As per the evaluation results in Section 5.4, energy sharing is able to significantly improve the max-min rate.
5.5. Conclusion

Figure 5.19: The value of $R_D$, $T_D$, $R_D$ and $T_D$ with varying $S$.

Figure 5.20: Running time with varying $S$. 
Further, $CHMM$ has similar performance with the formulated LP. In addition, the results also indicate that any increase in $|S|$, $P_T$, $\eta$ or $|V|$ has a positive impact on the max-min rate. However, more sinks reduces the max-min rate.
Chapter 6

Conclusion

Data gathering is a basic operation in WSNs. Existing works on data gathering usually aim to maximize throughput at the sink or to allocate rates fairly to sources. To date, these past works usually place additional relay nodes or optimize the sensing rate of sources to achieve their objective. However, a key problem is that nodes in these works have limited energy. Thus, researchers have begun to consider energy harvesting nodes. Unfortunately, nodes experience uncontrollable energy supply. Compared to ambient environment sources, WPT is able to provide a stable energy source. However, as per Chapter 2, there are limited works that employ WPT to recharge sensor nodes to optimize data gathering in WSNs. This thesis thus adds to the state-of-the-art by proposing and addressing new problems and solutions that consider using WPT to optimize data gathering in energy harvesting WSNs. Moreover, it investigates whether WPT is a viable approach to improve the amount of data gathered from a WSN.

This thesis proposes and studies three novel problems: ACP-MF, ACP-MM and RFES-MM. In particular, as per Chapter 3, ACP-MM has as its objective to maximize the throughput at one or more sinks by selecting a set of sensor nodes to be upgraded or supplemented with an AC. A key consideration is the finite number of ACs. Further, wireless interference is considered. To solve the problem in large
scale networks, Chapter 3 devises three algorithms; namely, *Path*, *LagOP* and *Tabu*. The results show that the performance of *Path* and *LagOP* is close to that of *Tabu*. Nevertheless, both *Path* and *LagOP* have a much smaller running time than *Tabu*. This chapter also explores the impact of the following parameters: number of nodes, number of sources, node degree and number of ACs. The results show that all parameters except number of nodes have a positive influence on throughput.

The second problem is called ACP-MM. The aim is similar to ACP-MM, but the objective is to maximize the minimum sensing rate. Chapter 4 models the problem as a MILP and proposes two novel heuristic algorithms: *GND* and *OUED*. It then outlines some properties of *GND* and *OUED*. Evaluation results show that the average gap of *GND* and MILP is 0.66% in small networks. On the other hand, the average gap between *OUED* and *GND* is 2.13%. In large networks, *GND* and *OUED* have similar performance. However, the running time of *GND* is 91.91 times higher than that of *OUED*. In addition, Chapter 4 also studies the impact of the number of nodes, number of sources, node degree and number of ACs. The results show that the number of sources and nodes have a negative impact on the max-min rate. By contrast, node degree and number of ACs have a positive influence on the max-min rate.

Lastly, Chapter 5 studies the RFES-MM problem: determine the time duration used for energy transmission/reception and data transmission/reception at each sensor node. The objective is to maximize the minimum sensing rate. The RFES-MM problem is modelled as a LP subject to ten constraints. As a comparison, a heuristic algorithm called *CHMM* is proposed. Next, this chapter investigates the impact of the following factors: number of sources, transmit power, conversion efficiency, number of nodes and number of sinks. The results show that the performance of *CHMM* is close to that of the formulated LP in experiments with varying number of sources, transmit power, conversion efficiency or number of nodes. However, the gap between *CHMM* and the formulated LP rises when the number of sinks increases.

The aforementioned studies confirm that WPT helps improve the performance
of data gathering in WSNs. For example, Chapter 3 shows that deploying a finite number of ACs improves the max throughput by 412.14%. Further, in Chapter 4, the max-min rate increases by as much as 135.30% when ACs are deployed. Lastly, Chapter 5 shows that the max-min rate increases by 13.98% when nodes are able to share their energy. In addition, compared to solar energy harvesting, WPT provides a stable energy supply.

Using WPT in WSNs, however, has some limitations. In Chapter 3 and 4, this thesis employs magnetic resonant coupling as a WPT technology; notably, it has a high energy efficiency. However, one AC only charges one node each time. In addition, an AC must be deployed next to a sensor node and must be aligned correctly to ensure maximum power transfer. Thus, an accurate location algorithm is required to obtain the position of sensors. In Chapter 5, this thesis studies RF wireless charging. In contrast to magnetic resonant coupling, RF charging is able to simultaneously charge multiple nodes and has a longer energy transmission distance. However, RF charging has a high energy loss, meaning it can only be used by low power nodes. Alternatively, nodes need to be placed very near an energy transmitter before a high energy conversion efficiency is achieved.

With regard to future research, there are numerous directions. First, a key future work will be to consider random energy harvesting rates. That is, given some probability distribution of the energy harvesting rates at each node, solve the ACP-MF, ACP-MM and RFES-MM problem in this new setup. Second, in the RFES-MM problem, a key future work is to develop a distributed solution. In particular, sensor nodes first exchange their data queue length. Based on this queue length information, each sensor node determines the time used for energy and data. Third, Chapter 5 employs Friis’ equation to represent the received power. A possible future work is to consider the physical interference model or one where nodes use pilot symbols to obtain the channel state information.
Bibliography


[53] H. Nishimoto, Y. Kawahara, and T. Asami, “Prototype implementation of am-
bient rf energy harvesting wireless sensor networks,” in *IEEE Sensor*, (Kona, HI), Nov. 2010.


[61] N. Wang, Y. Zhu, W. Wei, J. Chen, S. Liu, P. Li, and Y. Wen, “One-to-
multipoint laser remote power supply system for wireless sensor networks,”

using a light beam,” January 2015.

[63] V. Talla, B. Kellogg, B. Ransford, S. Naderiparizi, S. Gollakota, and J. Smith,
“Powering the next billion devices with wi-fi,” in ACM CoNext, (Heidelberg,
Germany), Dec. 2015.

wirelessly-powered platform for sensing and computation,” in UbiComp, (Or-
ange County, CA), Sept. 2006.

[65] M. Soyturk and D. T. Altilar, “Reliable real-time data acquisition for rapidly
deployable mission-critical wireless sensor networks,” in INFOCOM Work-
shops, (Phoenix, AZ), Apr. 2008.

sensor networks in forest fire detection and monitoring,” Computers, Environ-


[68] R. Zhang and C. K. Ho, “MIMO broadcasting in simultaneous wireless infor-
mation and power transfer,” IEEE Transactions on Wireless Communications,

problems;” Management science, vol. 27, no. 1, pp. 1–18, 1981.

[70] Y. Peng, Z. Li, W. Zhang, and D. Qiao, “Prolonging sensor network lifetime
through wireless charging,” in IEEE RTSS, (San Diego, California, USA),
Nov. 2010.


C. Wang, J. Li, F. Ye, and Y. Yang, “Multi-vehicle coordination for wireless
energy replenishment in sensor networks,” in IEEE. Intl. Symp on Parallel and Distributed Processing, (Boston, Massachusetts USA), May 2013.


