PERFORMANCE ANALYSIS OF DATA RETRIEVAL IN WIRELESS SENSOR NETWORKS

Mihaela Mitici
PERFORMANCE ANALYSIS OF DATA RETRIEVAL IN WIRELESS SENSOR NETWORKS

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof. dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended
on Wednesday, the 18th November 2015, at 14:45

by

Mihaela Angelica Mitici

born on 10th April, 1988
in Ploiesti, Romania.
This dissertation has been approved by:
Prof. dr. R.J. Boucherie (promotor)
dr. ir. J. Goseling (co-promotor)
dr. ir. M. de Graaf (co-promotor)
The mere formulation of a problem is far more essential than its solution, which may be merely a matter of mathematical or experimental skills. To raise new questions, new possibilities, to regard old problems from a new angle, requires creative imagination and marks real advances in science.

Albert Einstein
Acknowledgments

There were four years of research, it was an amazing journey and I would like to thank several people who, in various ways, played an important role in the realization of this PhD thesis.

Firstly, I would like to thank my promoter Richard J. Boucherie and my co-promoters Maurits de Graaf and Jasper Goseling for their guidance and constant support during my PhD. Richard, thank you for steering my research in the right direction and keeping me focused throughout my PhD. Maurits, I am deeply grateful for all the time we spent discussing about my research, your patience and positive attitude. Jasper, I always greatly appreciated your valuable technical input, your keen eye for clarity and elegance in scientific writings, and, most importantly, your constant belief in me. I am very grateful for all your support.

This thesis would have been quite different without the help and support of Nico van Dijk and Jan-Kees van Ommeren. Mr. van Dijk, from being my master coordinator to writing together scientific papers during my PhD, you have always guided my steps, encouraged me and firmly reassured me that scientific value always prevails. Jan-Kees, thank you for keeping your office door open for my ‘small’ questions. Thank you for taking the time to discuss about my research ideas, I always valued your technical input. Mr. van Dijk, Jan-Kees, it was a pleasure to learn from you and work with you.

A special thank you to Nelly Litvak. Nelly, I am very grateful for encouraging me to look at problems from a new perspective when stuck in a proof, to try out new techniques and, most importantly, to keep confident about the outcome. I also value our life-related discussions, which I think will serve me well in the future.

I would also like to express my gratitude to all my colleagues at the Stochastic Operations Research (SOR) group for the warm research environment. Yanting, Pim, Anna, Berksan, Corine, Xinwei, Anne, Joost, Thyra, Judith, Arnoud, Werner, Maartje, Niek, Aleida, Ingeborg, Sem, Ruben, Kamiel, Jasper, Tom, Daniel, thank you for all the enjoyable time during coffee breaks, seminars or conferences. In particular, I would like to thank my officemates Yanting, Pim, Anna and Tom for the pleasant company. Yanting, Pim, thank you for your support and for constantly reminding me that every cloud has a silver lining.

Apart from doing research during my PhD, I also enjoyed being part of the P-NUT board (PhD Network of University of Twente). It was great to work with the P-NUTers to better inform, connect and represent our fellow PhDs. I also enjoyed having Lindyhop dance lessons with the Swing Out Loud team and taking gliding lessons at the Drienerlo Gliding Club (DZC).
A special thank you goes to all my friends in the Netherlands. Vera, Sahar, Jessica, Corina, Leila, Daniela, Olga, Aura, Ivana, Sarah, Laura thank you for the nice times we had over the last few years, for your constant help and encouragement. Because of you, Enschede felt like home. Gentlemen, Csaba, Matthias, Edo, Felix, Paolo, Gjis, Milos, Andrei, Wilbert, Peter, your company was equally enjoyable. A special thank you goes to my friends in Romania, Mariana, Miha and Alina, for their never-fading friendship, their hospitality and constant support.

Mulțumesc familiei mele pentru sprijinul necondiționat pe care mi l-au acordat pe parcursul stagiului de doctorat. Încurajările și atitudinea voastră pozitivă m-au motivat să îmi urmez visul. Ultimul gând este pentru Bijoy. Cu tine, totul are sens.

Mihaela Mitici

Enschede,
October 2015
## Contents

1 Introduction 1
   1.1 Sensors and Wireless Sensor Networks 2
   1.2 Applications of Wireless Sensor Networks 4
      1.2.1 Environmental Monitoring 5
      1.2.2 Military Applications 5
      1.2.3 Healthcare Applications 5
      1.2.4 Industrial Applications 6
      1.2.5 Smart Homes 6
   1.3 Sensor Networks: Characteristics and Challenges 6
      1.3.1 Reliability of Sensor Measurements 7
      1.3.2 Energy Consumption for Sensors 7
      1.3.3 Retrieval Time of Sensor Measurements 8
      1.3.4 Querying the Sensed Data 8
      1.3.5 Age of Sensor Measurements 9
      1.3.6 Storage for Sensor Measurements 9
      1.3.7 Security for Sensor Networks 9
      1.3.8 Sensor Clustering 10
      1.3.9 Scalability of Sensor Networks 10
   1.4 Models and Results 11
      1.4.1 Data Retrieval from Wireless Sensor Networks: Models and Main Results 11
      1.4.2 Data Retrieval from Caches: Model and Main Results 18
      1.4.3 Query Processing for Wireless Sensor Networks: Model and Main Results 21
   1.5 Outline of this thesis 23

2 Data Retrieval Time in Wireless Sensor Networks 27
   2.1 Introduction 27
      2.1.1 Data Reliability Model 29
   2.2 Part I: Data Retrieval Time - Asleep/Awake Sensor Cycles 29
      2.2.1 Model and Problem Statement 30
## CONTENTS

2.2.2 Analysis of Decentralized and Centralized Schedules  
2.2.3 Decentralized vs. Centralized Transmission Schedule  

2.3 Part II: Data Retrieval Time under Wireless Sensor Broadcasting  
2.3.1 Model and Problem Statement  
2.3.2 Preliminaries  
2.3.3 Behaviour of the System under Broadcasting  
2.3.4 Continuous-Time Markov Chain: General Multi-class Queue for Broadcasting  
2.3.5 Performance Measures  
2.3.6 Decentralized Broadcasting  
2.3.7 Analysis of Decentralized Broadcasting Schedule $D$  
2.3.8 Performance Measures - Decentralized Broadcasting  
2.3.9 Optimal Centralized Broadcasting  
2.3.10 Analysis of Centralized Broadcasting Schedule $O$  
2.3.11 Performance Measures - Centralized Broadcasting  
2.3.12 Decentralized vs. Centralized Broadcasting  

2.4 Conclusions  
2.5 Appendix  

3 Energy-Efficient Data Retrieval with Time Constraints  
3.1 Introduction  
3.2 Model and Problem Statement  
3.3 Energy Consumption and Tight Time Constraints: Greedy Schedule  
3.3.1 Greedy Schedule  
3.3.2 Preliminaries  
3.3.3 Expected Energy Consumption under Greedy  
3.3.4 Maximum Expected Transmission Range under Greedy  
3.4 Spatial Poisson Process - Greedy Schedule  
3.4.1 Energy Consumption for Greedy: Spatial Poisson Process  
3.4.2 Expected Maximal Transmission Range for Greedy: Spatial Poisson Process  
3.5 Energy Consumption - General Time Constraints  
3.5.1 Stochastic Dynamic Programming  
3.5.2 Heuristic  
3.5.3 Offline Schedule  
3.6 Numerical results
CONTENTS

3.7 Conclusions .................................................. 84
3.8 Appendix ..................................................... 86
  3.8.1 Proof of Lemma 3.2 ................................. 86
  3.8.2 Proof of Lemma 3.3 ................................. 86
  3.8.3 Proof of Lemma 3.4 ................................. 90

4 Data Retrieval Time for Energy Harvesting Sensor Networks 93
  4.1 Introduction .............................................. 93
  4.2 Model and Problem Statement ......................... 96
  4.3 Analysis ................................................... 97
    4.3.1 A Single Sensor .................................. 98
    4.3.2 Retrieving a Reliable Estimate .................. 99
    4.3.3 Asymptotic Analysis of the Retrieval Time of an Esti-
         mate ............................................... 102
  4.4 Numerical Results ....................................... 105
  4.5 Conclusions ............................................. 107

5 Caching: Deployment vs. Data Retrieval Costs 109
  5.1 Introduction .............................................. 109
  5.2 Problem Statement ...................................... 112
  5.3 Analysis ................................................... 114
    5.3.1 Partitioning ....................................... 115
    5.3.2 Coding ............................................. 115
    5.3.3 Performance Comparison Partitioning - Coding .. 118
    5.3.4 Non-optimal Coding vs. Partitioning ............ 119
  5.4 Conclusions ............................................. 120

6 Query Processing in Wireless Sensor Networks 121
  6.1 Introduction .............................................. 121
  6.2 Model Description ...................................... 124
  6.3 Continuous - Time Markov Decision Process with a Drift . 125
  6.4 Exponentially Uniformized Markov Decision Process ..... 128
  6.5 Discrete Time and Discrete Space Markov Decision Problem . 130
    6.5.1 Standard Stochastic Dynamic Programming ..... 132
  6.6 Query Assignment Heuristics ............................ 133
  6.7 Numerical Results for the Optimal Query Assignment
      Policy ................................................ 134
  6.8 Simulation Results ...................................... 136
    6.8.1 The Influence of the Freshness Threshold ....... 136
    6.8.2 The Influence of the Job Arrival Rates ......... 137
## CONTENTS

6.8.3 The Influence of the Service Rate .................................. 140  
6.9 Discussion ......................................................................... 141  
6.10 Conclusions ..................................................................... 141  
6.11 Appendix ........................................................................ 143  
   6.11.1 Proof of Theorem 6.1 ............................................. 143  
   6.11.2 Proof of Theorem 6.2 ............................................. 145  

7 Conclusions .......................................................................... 151  
   7.1 Contributions and Concluding Remarks .......................... 152  
   7.2 Future work ................................................................... 154  

Bibliography ........................................................................... 159  

Summary .................................................................................. 169  

Samenvatting .......................................................................... 171
CHAPTER 1

Introduction

Wireless sensor networks are currently revolutionizing the way we live, work, and interact with the surrounding environment. Due to their ease of deployment, cost-effectiveness and versatile functionality, sensors are employed in a wide range of areas such as environmental monitoring, control of industrial equipment, surveillance and public safety, and smart homes. Sensors provide, for instance, timely alarms in case of earthquakes and floods. Sensors are also used to monitor battlefields and track the movement of vehicles and forces. Deployed in smart buildings, sensors monitor the indoor heating and ventilation for comfortable, safe and energy-aware living and working environments.

While providing unprecedented opportunities for a variety of applications, current sensor networks face several challenges. Some of these sensor challenges are application-specific. For instance, fire-alarm applications aim to retrieving data about possible fire outbreaks in a timely manner, to allow for intervention, if needed. It is, therefore, important to develop mechanisms which ensure that data retrieval meets time constraints. Some sensor challenges are generic. For instance, it is often the case that the reliability of sensor measurements is influenced by factors such as the hardware characteristics of the sensors, the position of the sensors relative to the monitored phenomenon/object, the characteristics of the environment where the sensors are deployed. To acquire reliable information about an area monitored by sensors, it is often needed that several sensor measurements are aggregated. It is, thus, important to develop data aggregation techniques that increase data reliability. Another challenge for most current sensors is the fact that their functionality largely depends on the energy stored in their batteries. Therefore, it is important to investigate mechanisms which ensure an efficient use of sensor energy. Among other generic sensor challenges, we mention security threats such as eavesdropping attacks, limited bandwidth of the transmission medium and hardware failures, especially when the sensors are deployed in remote and hostile areas.
In this thesis, several aspects of wireless sensor networks are addressed. We focus on the analysis of the time needed to retrieve reliable data from a wireless sensor network. Moreover, we develop energy-efficient sensor transmission schedules that support the retrieval of reliable data from wireless sensor networks. We also analyze aspects such as distributed data storage techniques and query processing for wireless sensor networks. The performance of the systems considered in this thesis is analyzed numerically and analytically by means of various techniques such as the theory of stochastic processes and queueing, combinatorics, stochastic dynamic programming, Monte Carlo simulations. The analysis of the systems considered in this thesis provides insights into the impact of the characteristics of sensor networks (for instance, the deployment of the sensors, the transmission schedules, sleep/awake sensor cycles) on performance metrics such as the retrieval time of reliable data from wireless sensors, the energy consumption of the sensors, the load of the network. These insights provide a formal, theoretical support for the design, implementation and operation of wireless sensor network applications related to the retrieval of reliable data, query-based sensing and data storage.

The remainder of this chapter is structured as follows. In Section 1.1 we introduce sensors and wireless sensor networks. In Section 1.2 we discuss several applications for wireless sensor networks. In Section 1.3 we introduce several characteristics and challenges of wireless sensor networks. We state the mathematical models employed in this thesis, as well as a summary of the main results in Section 1.4. In Section 1.5 we outline the remaining chapters of this thesis and highlight its contributions.

1.1 Sensors and Wireless Sensor Networks

A sensor node is an electronic device with integrated sensing, data processing and communication capabilities. In the remainder of this thesis, a sensor node is referred to as a sensor. Figure 1.1 shows two examples of sensors equipped with batteries.

The main components of a sensor are the wireless sensing circuitry, the processing and storage unit, a wireless radio transceiver and a power unit. Figure 1.2 shows the basic architecture of a sensor. The sensing circuitry measures parameters of the environment surrounding the sensor such as temperature, CO$_2$ level, humidity, sound, light intensity or atmospheric pressure. It can also detect objects and can track their movement. The measurements are further processed using a micro-processor or a digital signal
1.1 Sensors and Wireless Sensor Networks

processor. After processing, a sensor uses wireless radio transceivers, usually with a single omni-directional antenna, to transmit its measurements to a sink.

![Image of sensor nodes equipped with batteries.](image)

The size of a sensor is, in general, in the order of a few centimeters. Due to their limited size, as well as due to cost constraints, most sensors are powered by small batteries, which are often hard or even impossible to replace if the sensors are deployed in remote, and possibly hostile, areas.

![Diagram of sensor architecture.](image)

Typically, a wireless sensor network is a collection of sensors which are deployed in an area of interest [3, 114]. The measurements of the sensors are usually transmitted to a sink/base station. Figure 1.3 illustrates a basic architecture for wireless sensor networks.

The deployment of the sensors is mainly deterministic or ad-hoc. When deterministically deployed, the sensors are manually placed at specific places in the area of interest and the sensor measurements are transmitted to a sink via predefined paths. Sensors can also be randomly deployed in an area
of interest and have to autonomously configure themselves into a network. This is the case, for instance, of sensor networks deployed in remote areas such as forests [11]. The number of deployed sensors is application specific, ranging from a few dozens in the case of, for instance, indoor climate monitoring, to a few thousands in the case of habitat or battlefield monitoring.

In most sensor networks, sensor measurements are transmitted via a wireless medium to a sink. The transmission of the sensor measurements is performed in a many-to-one, up-link fashion. Successfully received sensor measurements are collected by a sink for further processing and/or storage.

![Figure 1.3: Basic Architecture for a Wireless Sensor Network.](image)

The topology of the sensor networks often changes over time. One of the reasons is that, to conserve energy, sensors follow a sleep/awake cycle. When asleep, the radio and possibly other components of the sensors are turned off and, thus, the sensors are partially or not at all functional. The topology of the network can also change due to battery failure or environmental factors. This is especially the case of sensors deployed in hostile environments, where they are prone to physical damages.

### 1.2 Applications of Wireless Sensor Networks

Wireless sensor networks are currently employed in a variety of fields (see, for instance, [3, 17, 20, 36, 54, 58, 63, 66, 114] and the references therein). Below are a few examples of applications of wireless sensor networks, categorized into environmental monitoring, military applications, health applications, industrial applications and intelligent home applications.
1.2 Applications of Wireless Sensor Networks

1.2.1 Environmental Monitoring

Environmental monitoring is one of the most frequently encountered applications of wireless sensor networks. Atmospheric conditions, geo-physical contexts, chemical conditions and detection of objects are a few of the parameters measured by the sensors. Below are two examples of environmental monitoring applications.

- Habitat Monitoring

Sensors are often used to measure environmental conditions in wild habitats such as forest, remote islands [66]. Such sensor measurements enable end-users to remotely monitor, observe and control these habitats.

- Disaster Detection

Sensors can be randomly and densely deployed in forests or other eco-systems to detect, for instance, fires [11], floods [19] or seismic activity [98]. In forests, sensors are able to detect fire outbreaks and communicate the location of the fire to end-users. Sensors are particularly suitable for flood or earthquake detection since they can be randomly deployed in an area of interest and can function in hostile environments, without any infrastructure and little or no coordination.

1.2.2 Military Applications

Wireless sensor networks are becoming an important part of the military command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting systems [3,43]. Sensors are generally deployed in battlefields to detect, gather information and classify vehicles and forces and to track their movement. Sensors are also deployed for remote sensing of nuclear or chemical attacks.

1.2.3 Healthcare Applications

Sensors are increasingly used to monitor vital signs of patients such as heart rate or blood pressure, allowing remote, non-invasive monitoring [17]. This enables the patients to be aware of their health status. In addition, healthcare personnel can be alerted in case of a medical emergency [54] and, thus, medical care can be timely provided. Deployed in patients’ homes, sensors can monitor the behavior of the patients, whether they fall or require immediate medical attention [17,73].
1.2.4 Industrial Applications

In industry, wireless sensor networks enable manufacturing monitoring, industrial automation or monitoring and control of industrial equipment. For instance, sensors can be used to control the production process at an assembly line in a factory or can detect the status of the industrial equipment [97] and trigger alerts in case of equipment failure. The fact that the sensors are of small size makes it easier to place them in industrial machines, which otherwise would be difficult to monitor.

1.2.5 Smart Homes

The term ‘smart home’ is commonly used for a residence equipped with technology that allows monitoring of the living environment and of its inhabitants. Wireless sensor networks are increasingly deployed in houses with the goal of providing convenience, improved security and to save energy. Smart homes are equipped with embedded devices such that the functionality of the conventional appliances is enhanced. For instance, the sensors can detect the presence of humans in a room and adjust the light intensity accordingly. Occupancy detection can be used to save energy by automatically turning off the house’s heating and ventilation system if nobody is present in the house [24, 32], or to detect intruders [49]. Sensors are also used to monitor the climate inside smart houses (for instance, temperature, CO₂ level, humidity). The measurements are further sent to a remote center, which enables end-users to remotely access the data [13].

1.3 Sensor Networks: Characteristics and Challenges

Wireless sensor networks distinguish themselves from the traditional wireless communication networks such as mobile ad hoc networks (MANET) or cellular systems, through unique characteristics such as limited computational, storage and battery capabilities. Such features pose challenges for the development and deployment of wireless sensor networks.

Below we list several characteristics and challenges of wireless sensor networks. Some of these are addressed in this thesis. For instance, one of the aspects considered in this thesis is the time needed to retrieve sensor measurements from a wireless sensor network and the corresponding energy consumption. We also address the aspect of processing queries on sensed data, focusing on the freshness of the data provided to queries, on the response time of queries, as well as on storage possibilities for sensed
1.3 Sensor Networks: Characteristics and Challenges

data, either at the sensors or at central databases. Sensor clustering, security and scalability of wireless sensor networks are not addressed in this thesis.

1.3.1 Reliability of Sensor Measurements

The reliability of a measurement from an individual sensor, i.e., the value accuracy of a sensor reading, is often influenced by factors such as the position of the sensors relative to the monitored phenomenon/object, hardware features of sensors and the characteristics of the area where the sensors are deployed. Thus, it is often the case that the measurement of a single sensor is insufficient to obtain reliable information about the monitored area. To increase the reliability of the sensed data, measurements from several sensors are usually fused or aggregated. Data fusion techniques combine data from multiple sensors to obtain improved data accuracy (see, for instance, [16, 86, 100–102]).

The challenge is to develop techniques which support the aggregation of sensor measurements to obtain increased data reliability. In particular, these techniques should specify, for instance, which sensor measurements should be aggregated and should ensure that aggregates are available within a time window that is appropriate for the applications/end-users which request sensor measurements.

1.3.2 Energy Consumption for Sensors

A stringent constraint for wireless sensors is their limited energy availability. Energy is essential for sensors to sense the environment, process the information and communicate the sensed data.

Recent advances in low-energy circuits have enabled the development of low-energy hardware components for sensors [25, 82]. To further reduce the energy consumption, dynamic energy management techniques [42, 82, 88, 106] such as enabling sensors to enter a sleep mode if no events of interest occur or when dictated by an internal timer, are often used. It is thus important to develop energy-efficient medium access control (MAC) protocols to specify the time a sensor is awake/asleep so that an energy-driven communication is enabled in the sensor network.

Energy-efficient transmission schedules for sensors can further reduce the energy consumption. In most sensor networks, a central scheduler is responsible for scheduling the sensors for transmission. Often the challenge is to develop sensor transmission schedules or routing protocols to minimize energy consumption, as well as to meet certain time constraints. Ex-
tensive surveys on energy consumption in wireless sensor networks are, for instance, [9, 52].

Recent technological advances have enabled sensors to harvest energy from the environment. Energy harvesting technologies are, for instance, solar energy or mechanical energy from vibrations [5, 79, 84]. Such renewable energy resources have the potential to enhance the lifetime of sensors and to improve the overall performance of the network. However, harvested energy typically varies in time in a non-deterministic manner. The challenge lies in matching the energy availability at the sensors, which depends on the energy harvesting process, with the energy consumption of the sensors, and, in particular, with the energy consumed for data transmission.

1.3.3 Retrieval Time of Sensor Measurements

The retrieval time of sensor measurements specifies the period of time within which measurements are acquired by a sink or an end-user from the sensors. The retrieval time of sensor measurements is a frequently encountered Quality of Service (QoS) requirement for sensor networks, especially when a sink/end-user wants to obtain sensor measurements within strict time constraints.

The retrieval time of sensor measurements is directly influenced by the sensor transmission schedules, clustering and data aggregation protocols. The challenge is to develop transmission schedules that enable the retrieval of meaningful data about the monitored area while meeting time constraints.

1.3.4 Querying the Sensed Data

To retrieve sensor measurements, end-users query, via the Internet or using web portals, the sensed data. These queries are addressed to the sensor network or to central databases, where sensed data is stored. Queries are generally formulated using the SQL language, which uses SELECT-FROM-WHERE blocks to specify which type of data is required. For instance, (SELECT temp FROM sensors SAMPLE PERIOD 1s) is a query specifying that each sensor should report its temperature reading every second.

There are three main types of queries: event-driven, on-demand and persistent. In the case of event-driven queries, the transmission of data is triggered when an event occurs. On-demand queries are generated by end-users/sink. In the case of persistent queries, sensors send data periodically.

Depending on the applications and on the design of the sensor network, different time constraints are imposed on the retrieval of the data. One often
encountered challenge is to balance the retrieval time of sensor measurements provided to queries with the load of the sensor network.

### 1.3.5 Age of Sensor Measurements

The age of a sensor measurement describes how stale or old a measurement is with respect to the source [21].

The age of sensor measurements is possibly high for measurements stored in central databases in comparison to measurements retrieved directly from the sensor network. Stored measurements are often used to describe the historical evolution of the sensed parameters (for instance, the temperature values in a room over the last week). For real-time sensor applications, where the aim is to characterize the monitored areas with high fidelity, it is desired that the age of the measurements is low. In this case, the measurements are usually retrieved directly from the sensor network. The challenge is to provide applications/end-users with relevant sensor measurements while meeting constraints regarding the retrieval time of sensor measurements and sensor energy consumption.

### 1.3.6 Storage for Sensor Measurements

Sensor measurements are stored within the network, in situ, or externally, in central databases.

When stored externally, at the sink or at central databases, the sensor measurements are available for querying at a later time, without any additional need for sensor transmissions. When stored locally, at the sensor that generated the data, no additional energy is consumed by the sensor to transmit the data to external storage locations.

The challenge is to develop storage techniques that can meet the requirements of applications/end-users with respect to the retrieval time of sensor measurements and freshness of the measurements, while taking into account the storage capabilities of the sensors or databases.

### 1.3.7 Security for Sensor Networks

Since sensor networks usually operate unattended, wireless sensor networks are subject to the risk of physical attacks or unauthorized listening of the communication channel. Similar to traditional networks, most sensor network applications use authentication and cryptography to enhance the se-
curity level. Incorporating cryptography into the sensor network [80], however, is difficult due to the limited computational capabilities of the sensors.

The challenge is to provide security models tailored for sensor networks and, in particular, to develop novel cryptographic approaches for sensors with limited processing capabilities.

1.3.8 Sensor Clustering

Data transmission within the sensor network is often based on clustering techniques, where sensors dynamically form groups, and transmissions are mainly performed within the groups. As an example, an often used mechanism is for each group of sensors to select a sensor to be a cluster head. A cluster head collects the data from sensors from his group. After collecting sufficient data, a cluster head further relays the data to the sink. As examples, cluster-based sensor networks are assumed in [45], where a clustering-based protocol LEACH (Low-Energy Adaptive Clustering Hierarchy) randomly assigns sensors to be cluster-heads to evenly distribute the energy consumption among sensors. In [61], sensors also take turns to transmit to the sink using a chain-based schedule.

The challenge is to develop clustering schemes that guarantee the retrieval of desired data while meeting energy and time constraints. In addition, it is important to optimize the size of the clusters and the selection of the cluster heads so that energy consumption is evenly spread across the sensors and, hence, the lifetime of the network is prolonged.

1.3.9 Scalability of Sensor Networks

Scalability of a wireless sensor network refers to the ability of a network to accommodate an increasing number of sensors, a large volume of sensed data, as well as an increasing number of applications/end-users that query the sensed data. For a sensor network to be scalable, adding extra sensors or improving the characteristics of the current sensors (memory, processor power, low power consumption) should result in an increased (or, at least, the same) overall performance of the network.

The challenge is to provide scalable sensor deployment frameworks that support adequate coverage of an area of interest while meeting Quality-of-Service (QoS) and energy requirements.
1.4 Models and Results

The purpose of this section is to provide an overview of the mathematical models considered in this thesis and to introduce the main results, which will be rigorously proven in the rest of the chapters. The models and corresponding results are divided into three parts. Firstly, we introduce a general model for data retrieval from wireless sensor networks. Various aspects of this model are considered in Chapters 2, 3, 4, where we determine the time to retrieve data from the network, as well as the energy consumed by the sensors that provide this data. Secondly, we introduce a model for wireless caches which store data in a distributed fashion. This is the basis of Chapter 5. In this setting, we analyze the relation between the deployment of the caches and the process of retrieving data from these caches. Thirdly, we introduce a model for query processing for wireless sensor networks, which is the basis of Chapter 6. We provide a framework that allows for optimal query processing with respect to the response time of queries, as well as the age (freshness) of the data provided to queries.

1.4.1 Data Retrieval from Wireless Sensor Networks: Models and Main Results

Consider the situation where sensors are placed in the plane according to some spatial random process (see Figure 1.4). Each sensor has a measurement of an attribute $\theta$ (for instance, temperature, position of an object). These measurements are subject to additive normally distributed noise with zero mean and variance $\sigma^2$.

Clients arrive according to a Poisson process with rate $\lambda_a$ and are located at random places in the plane. Clients can be viewed as, for instance, mobile sinks or users with a mobile device that enter the area where the sensors are deployed and retrieve sensor measurements.

Each client is interested in retrieving from the sensor network $s \geq 1$ measurements, $s \in \mathbb{N}$, from distinct sensors. As soon as a client retrieves $s$ measurements, this client applies data fusion techniques in order to obtain an aggregate (for instance, average temperature, coordinates of an object).

Sensors transmit their measurements to the clients according to some transmission schedule. In particular, we focus on centralized and decentralized transmission schedules. We consider centralized transmission schedules where a scheduler specifies which sensors have to transmit and at which time. We also consider decentralized transmission schedules where sensors
transmit their measurements at random, independently of the other sensors.

These transmission schedules take into account the sleep/awake state of a sensor since sensors are able to transmit only if they are awake (if asleep, the radio component is turned off and no message can be received or transmitted). Moreover, a sensor is able to transmit its measurement only if it has sufficient energy for a transmission. Each sensor is equipped with a battery with finite capacity. At an exponential rate $\lambda_e$, the battery of an arbitrary sensor is charged with some amount of energy (possibly using a technology that enables sensors to harvest energy from the environment).

When a sensor $i$ transmits a measurement to a client $j$ and the distance between this sensor and this client is $\delta_{i,j}$, the energy consumed by sensor $i$ is $\delta_{i,j}^a$, where $a \geq 1$ is a fixed parameter denoting, for instance, the path-loss exponent of the transmission environment.

We are interested in determining metrics such as the time for a client to retrieve $s$ measurements from distinct sensors, based on which an aggregate can be computed, and the energy consumption of the sensors that provide this client with measurements.

The model above introduces a general, yet complex wireless sensor network. In this thesis, we consider more specific model formulations, which extend some of the assumptions of the general model above, while relaxing others. Below we introduce four such specific models (Models a), b), c) and d)), which are the basis of Chapters 2, 3, 4, and state the corresponding main results. Table 1.1 shows which aspects are considered in each model. Some aspects are addressed in more than one model. For instance, all models consider decentralized sensor transmission schedules, but only Model b) and c) consider the energy consumption of the sensors.
1.4 Models and Results

<table>
<thead>
<tr>
<th>Model for Data Retrieval</th>
<th>a)</th>
<th>b)</th>
<th>c)</th>
<th>d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous-time model</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Slotted-time model</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Retrieval time of (s) sensor measurements</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Energy consumption</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Random deployment of sensors in the plane</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Multiple clients</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Centralized sensor transmission</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Decentralized sensor transmission</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Sleep/Awake sensor cycle</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Energy harvesting</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Finite sensor battery</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

Table 1.1: Characteristics of Models a), b), c) and d) on data retrieval for wireless sensor networks.

a) Model on Data Retrieval with Sleep/Awake Sensor Cycles and Main Results

Consider a time-slotted model where a single client is interested in retrieving \(s \geq 1\) measurements from distinct sensors of a network of \(N\) sensors, \(N \geq s\). Every time slot a sensor is awake with probability \(0 < p < 1\) (and asleep with probability \(1 - p\)). Sensors transmit their measurement to the client according to some centralized schedule \((C)\), which specifies which sensors have to transmit and at which time, or a decentralized schedule \((R)\), where sensors transmit independently, at random. Both schedules take into account the sleep/awake state of the sensors. The duration of a measurement transmission is one slot.

We are interested in the time \((W_s)\) the client retrieves \(s\) measurements from distinct sensors under the \(C\) and \(R\) schedules, \(s \in \mathbb{N}, 1 \leq s \leq N\).

The main result obtained for the model above is that despite the contrasting settings of the centralized and decentralized schedules, we show that the retrieval time of \(s\) measurements in the decentralized framework is a constant factor \(e\) approximation to the centralized framework. This result, summarized in the theorem below, is rigorously proven in Chapter 2, Part I.
Theorem 1.1. As the size of the sensor network $N \to \infty$, the performance gap, with respect to the retrieval time of $s$ sensor measurements, between the decentralized $R$ sensor transmission schedule and the centralized $C$ sensor transmission schedule is a constant factor $e$. Formally,

$$\lim_{N \to \infty} \frac{E[W^R_s]}{E[W^C_s]} = e.$$ 

The results show that, for sufficiently large networks, even when the sensor network functions according to a simple, decentralized protocol, with no coordination between the sensors and no information about the sleep/awake mode of the sensors, the system still performs well in terms of the retrieval time of a specific number of measurements, in comparison to the centralized setting, which assumes sensor coordination and known sleep/awake sensor mode, aspects which are often difficult to achieve in real-life situations.

b) Model on Data Retrieval with Time Constraints and Sensor Energy Consumption and Main Results

Similar to the previous model, we consider a time-slotted model where every time slot a sensor is awake with probability $0 < p < 1$ (and asleep with probability $q = 1 - p$). A single client, at a random location in the plane, is interested in retrieving exactly $s$ measurements from distinct sensors. When awake, an arbitrary sensor transmits its measurement according to some schedule.

In addition to the previous model, in this setting we also consider the geometry of the sensor deployment. More precisely, we consider sensors located in the plane according to a spatial Poisson process with intensity $\lambda$.

An additional aspect that we take into account is the energy that the sensors consume when they transmit measurements to the client. We assume that the energy consumption of sensor $i$ transmitting to the client is $\delta^a_i$, where $\delta_i$ denotes the distance between sensor $i$ and the client and $a \geq 1$ is a fixed parameter which denotes, for instance, the path-loss exponent of the transmission environment.

Lastly, we assume that the client has only a predefined time period of $t \geq s$ time slots within which he has to retrieve exactly $s$ measurements from distinct sensors.

We are interested in developing transmission schedules that minimize the energy consumption of the sensors while ensuring that exactly $s$ measurements are retrieved by the client within $t \geq s$ time slots (Figure 1.5).
The main results obtained for the above model, which are rigorously proven in Chapter 3, are the following.

i) We formulate a simple, intuitively appealing, sensor transmission schedule (Greedy) that schedules every time slot the closest, awake sensor, which has not been scheduled previously. We provide a closed-form expression for the energy consumption of this schedule, as stated in the theorem below.

**Theorem 1.2.** Under Greedy schedule, the total expected energy consumption for the transmitting sensors is:

\[
C_s \Gamma(a + 1) \frac{1}{\lambda \pi^{a/2}} \sum_{j=1}^{s} \frac{(-1)^{j+1} q^{(s-j)(s-j+1)/2}}{q^j \prod_{i=1}^{s-j} (1 - q^i)} \frac{q^j}{(1 - q^j)^{a+1}},
\]

where \( C_s = p^s q^{s(s+1)/2} \prod_{m=0}^{s-1} \sum_{v=0}^{m} q^v \) and \( \Gamma(\cdot) \) is the gamma function defined as

\[
\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt \quad \text{for} \quad a > 0.
\]

ii) We formulate a stochastic dynamic programming model to determine a sensor transmission schedule with general time constraints such that exactly \( s \) measurements from distinct sensors are retrieved with minimum energy consumption for the sensors. Moreover, for the case when \( t = s \) time...
slots are allocated to retrieve exactly $s$ measurements, we argue that Greedy is an optimal, online schedule.

The results provide a formal framework that specifies which sensors should transmit their measurements and when, such that the client retrieves exactly $s$ sensor measurements within a predefined time and with minimal energy consumption for the sensors. Moreover, the results provide insight into the impact of the time allocated for sensor measurement retrieval, the number of measurements to be retrieved and the energy consumption of the transmitting sensors.

c) Model on Data Retrieval and Sensor Broadcasting and Main Results

In contrast to the model above, in this model we consider a continuous-time framework where multiple clients arrive at a network of $N$ sensors, $N \geq s$, according to a Poisson process with rate $\lambda_a$. We do not consider here the spatial deployment of the sensors. Similar to the models above, each client is interested in retrieving exactly $s$ measurements from distinct sensors. Based on these measurements, each client computes an aggregate. We further assume that at an exponential rate $\mu$, sensors broadcast their measurements according to an optimal, centralized ($O$) or decentralized ($D$) broadcasting schedule. Figure 1.6 shows an example of a realization of some arbitrary sensor broadcasting schedule.

We are interested in the time ($W_s$) in which the client retrieves $s$ measurements from the sensor network under the $O$ and $D$ broadcasting schedules.

The main results obtained for the above model are based on a novel multi-class queueuing system which we introduce in Chapter 2, Part II. We
1.4 Models and Results
determine the steady-state distribution of the system and consequently, are able to determine performance metrics such as the time for a client to retrieve \( s \) sensor measurements, the load of the system and the busy period of the system. The theorem below states one of the results of Chapter 2, Part II.

**Theorem 1.3.** The expected retrieval time for an arbitrary client under the \( D \) and \( O \) broadcasting schedule, respectively, is:

\[
\begin{align*}
i) \quad \mathbb{E}[W_s^D] &= \frac{s-1}{\mu(1-i/N)}, \\
ii) \quad \mathbb{E}[W_s^O] &= \frac{s}{\mu}.
\end{align*}
\]

The results quantify the performance of the system under two contrasting, centralized and decentralized sensor broadcasting schedules. These results are derived from a simple, continuous-time model that can accommodate various broadcasting schedules, which emphasizes the generality of the model.

d) Model on Data Retrieval and Energy Harvesting and Main Results

Similar to the previous model, we consider a continuous-time system where clients arrive at a network of \( N \) sensors, \( N \geq s \), according to a Poisson process with rate \( \lambda_a \). As in all previous models, each client is interested in retrieving \( s \) measurements from distinct sensors of the network.

In addition, in this model we consider sensors that harvest energy from the environment in order to charge their batteries. Let \( b \in \mathbb{N} \) be the energy level of an arbitrary sensor in the network, where \( 0 \leq b \leq B < \infty \). At an exponential rate \( \lambda_e \), an arbitrary sensor harvests one unit of energy from the environment. If \( b = B \) and a new unit of energy is harvested, then this harvested energy is discarded. The energy available at a random sensor is modeled as a Birth-and-Death Markov process (see Figure 1.7).

If a random sensor has at least one unit of energy, then at an exponential rate \( \mu/N \) this sensor broadcasts its measurement, independently of the other sensors. Upon a broadcast, the total energy available at the broadcasting sensor decreases by one energy unit.

We are interested in the time \((W_s)\) in which the client retrieves \( s \) measurements from distinct sensors of the network.

The main results obtained for the above model, which are rigorously proven in Chapter 4, consist of providing the distribution of the time \((W_s)\) for a client to retrieve \( s \) measurements from distinct sensors. Consequently,
we are able to derive the expected retrieval time of \( s \) measurements for a client, which is formally stated in the theorem below.

**Theorem 1.4.** The expected time for an arbitrary client to retrieve \( s \) measurements from distinct sensors is:

\[
\mathbb{E}[W_s] = \sum_{j=0}^{s-1} \binom{N}{j} \sum_{k=0}^{j} \binom{j}{k} (-1)^{j-k} \sum_{v=0}^{N-k} \binom{N-k}{v} \cdot \frac{\omega^v (1 - \omega)^{N-k-v}}{(\lambda_e (N-k-v)) + \mu/N},
\]

where \( \omega = 1 - \nu(0) \frac{\mu}{N - \lambda_e} \) and \( \nu(0) \) is the steady-state probability of an arbitrary sensor having no energy stored in its battery.

The results illustrate the impact of the energy harvesting process assumed in this model on the retrieval time of a fixed number of sensor measurements. These results are obtained for the setting where no coordination is needed between sensors for transmission or energy harvesting, which is often the case of sensors deployed in remote areas that need to function unattended. Moreover, we show that these results also hold for Model c) when the rate of energy harvesting is asymptotically large.

### 1.4.2 Data Retrieval from Caches: Model and Main Results

In this model we consider wireless caches, placed in the plane according to a spatial Poisson process with intensity \( \lambda_d \). These caches store parts of a *data file* in a distributed fashion. A client, at a random location in the plane, is interested in retrieving sufficient data from the caches to acquire the complete data file. Although this model does not consider a wireless sensor network, as in the previous models, the main objective remains the same, i.e., to retrieve data from a predefined number of distinct devices. In the previous
models, a client was interested in retrieving measurements from the wireless sensor network to compute an aggregate. Following a similar setting, in the current model a client is interested in retrieving parts of a file from wireless caches to obtain the complete file.

We further assume that the file consists of \( n \) parts. A cache stores \( n/k \) parts (we assume that \( n \) is divisible by \( k \)) according to network coding and partitioning storage techniques.

Under network coding (see, for instance, [35]) caches store random linear combinations of parts of the file. We assume that any two linear combinations are independent. To obtain the complete file, the client retrieves data from any \( k \) caches (see Figure 1.8a for an example on data caching with network coding).

Under partitioning, replicas of parts of the file are stored at the caches. A cache is assigned at random a replica of a file part. If a specific part is not stored in at least one cache, then we reallocate the file at the caches. A client retrieves data from \( k \) caches which store replicas of distinct parts of the file. Figure 1.8b shows an example of caching with partitioning.

Again, the similarity between the models on data retrieval from a wireless sensor network and the current model is that data needs to be retrieved from a fixed number of devices (\( s \) measurements from distinct sensors, in the case of the previous models on sensor networks, whereas \( k \) file parts from distinct caches, in the case of the current model).

We further consider two conflicting objective functions.

Firstly, we consider the cost of retrieving the data from caches, denoted by \( C_r(k, \lambda_d) \), which is an increasing function of the distance between the client and the cache providing the data.
Secondly, we consider the deployment cost $C_d(k, \lambda_d)$ of the caches, which is an increasing function of the intensity according to which caches are placed in the plane.

Finally, we consider the multi-objective optimization problem for $1 \leq k \leq n, \lambda_d > 0$ and storage technique $B \in \{\text{Code, Part}\}$:

$$\begin{align*}
\min & \quad C^B_r(k, \lambda_d) \\
\min & \quad C_d(k, \lambda_d)
\end{align*}$$

The main results obtained for the above model consist of an exact characterizations of the Pareto front (see, for instance, [22]) of the retrieval and deployment cost functions under coding and partitioning (see Figure 1.9). We also show that the Pareto front under coding is always lower than the Pareto front under partitioning. These results are formally introduced below and are rigorously proven in Chapter 5.

![Figure 1.9: The Pareto front for coding and partitioning, $2\alpha \geq 1$ is a fixed parameter denoting the path-loss exponent of the transmission environment. The parameters over which we optimize are $\lambda_d$ and $k$.](image)

**Theorem 1.5.** The Pareto front of the partitioning and coding storage technique, respectively, is described by the following set of points:

1. $\left\{(x_{\text{Part}}, y_{\text{Part}}) \mid x_{\text{Part}} > 0, y_{\text{Part}} = \frac{\Gamma(\alpha + 1)}{(\pi x_{\text{Part}})^\alpha}\right\}$.
2. $\left\{(x_{\text{Code}}, y_{\text{Code}}) \mid x_{\text{Code}} > 0, y_{\text{Code}} = \frac{\Gamma(\alpha + 1 + n)}{(\alpha + 1)\Gamma(n)(\pi x_{\text{Code}})^{\alpha n + 1}}\right\}$.
Theorem 1.6. There exist no values for the parameters $\lambda_{d_{\text{Part}}}$, $\lambda_{d_{\text{Code}}}$ and $1 < k_{\text{Part}}, k_{\text{Code}} \leq n$ such that $C_{d_{\text{Part}}}(k_{\text{Part}}, \lambda_{d_{\text{Part}}}) \leq C_{d_{\text{Code}}}(k_{\text{Code}}, \lambda_{d_{\text{Code}}})$ and $C_{r_{\text{Part}}}(k_{\text{Part}}, \lambda_{d_{\text{Part}}}) \leq C_{r_{\text{Code}}}(k_{\text{Code}}, \lambda_{d_{\text{Code}}})$, with at least one of these inequalities holding strictly.

The results show to what extent one of the two objective functions (retrieval and deployment cost functions) can be improved at the expense of the other. Moreover, in this setting, we show and quantify the benefit of network coding in comparison to the partitioning storage technique.

1.4.3 Query Processing for Wireless Sensor Networks: Model and Main Results

In this mode, we consider the situation where queries about an attribute monitored by a sensor network are processed either with data retrieved directly from the sensor network (WSN) or from a central database (DB), where sensed data is periodically stored.

Processing queries at the WSN or at the DB poses advantages and disadvantages with respect to the response time of queries and the age (freshness) of the data provided to queries. If processed by the WSN, queries join a queue and wait to be processed. But the age of the data provided to these queries is negligible (the data has been recently generated by the sensors). If processed by the DB, the response time of the queries is negligible since data is already stored and available at the DB, but the age of this data is possibly large, depending on the time when the data was stored.

Figure 1.10 illustrates our model on query processing. Queries, generated at an exponential rate $\lambda_1$, are processed either by the WSN or by the DB. At an exponential rate $\lambda_2$, reports are generated and processed by the WSN only. After it is processed, a report updates the DB. The service requirement of a query or a report is exponentially distributed with parameter $\zeta$.

![Diagram](Image)
The WSN processes queries according to a Processor Sharing type of service. Thus, the response time of queries processed by the WSN depends on the load of the WSN. Moreover, we consider stored data to be fresh if the period between the time at which the data was stored at the DB and the time this data is provided to a query, is below a fixed threshold $T > 0$.

We are interested in query assignment strategies which trade-off between the freshness of the data provided to queries and the response time of queries. We formulate a Markov decision problem to determine an optimal query assignment strategy which specifies whether a new query is processed at the WSN or at the DB (see Figure 1.11). We employ a Markov process with the state space $S = \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0$, where $(i, j, A) \in S$ denotes a state with $i$ queries and $j$ reports waiting to be processed within the WSN, and $A$ the age of data stored at the DB.

The action is to assign an incoming query to the DB or to the WSN. The set of optimal actions taken by the controller when the system is in state $(i, j, A)$ and a new query arrives at the system determines an optimal query assignment policy. When in state $(i, j, A)$, a cost $i$ is incurred for the $i$ queries present at the WSN, plus an instantaneous cost $(A - T)^+$ if a query is assigned to the DB. The cost $i$ of having $i$ queries in the WSN indicates the response time of these queries (due to Processor Sharing type of service assumed for the WSN, the response time of the queries is an increasing function of the number of jobs being processed by the WSN). The cost $(A - T)^+$ upon a DB assignment penalizes the case when queries are provided with no longer fresh data. If $A > T$, this penalty increases linearly with the age $A$ of the data stored in the DB.

The main results following the above model, which are rigorously proven in Chapter 6, are as follows.

i) We formulate a discrete time and discrete state Markov process that characterizes the query assignment problem described above.

ii) We determine an optimal query assignment strategy, which specifies when it is optimal to assign an incoming query to the WSN or the DB, and compare the performance of this strategy with several assignment heuristics, derived from practice.

These results provide a formal framework for processing queries in an optimal manner with respect to the response time of queries and the quality of the data provided. The comparative analysis of our optimal query assignment policy and the heuristics proposed for this model provide a formal support for the design of query-based applications for sensor networks, where simple heuristics can be designed to perform close to the optimum.
Figure 1.11: Example of an optimal query assignment strategy, which specifies whether it is optimal to assign an incoming query to the DB or to the WSN, given that there are $i$ queries, $j$ reports in the system and the current age $A$ of the stored data; $\lambda_1 = 0.8, \lambda_2 = 0.5, \zeta = 1.8, T = 4, A = 50$, DB assignment (white), WSN assignment (blue). The optimal policy trades-off between the cost related to the response time of the queries and the cost related to the freshness of the data provided to queries.

1.5 Outline of this thesis

This thesis is structured as follows.

In Chapter 2, we analyze the time to retrieve a reliable estimate of an attribute monitored by a sensor network.

In Chapter 2, Part I, we consider the case when a single client is interested in retrieving a reliable estimate from a wireless sensor network. Centralized and decentralized sensor transmission schedules, as well as asleep/awake sensor cycles are considered. We determine the retrieval time of a reliable estimate in the decentralized and centralized framework. Moreover, we show that, as the size of the sensor network increases, the retrieval time under the decentralized schedule is a constant higher than under the centralized schedule.

In Chapter 2, Part II, we consider the case when multiple clients access the sensor network to retrieve a reliable estimate. We formulate a novel, multi-class queueing model that accommodates various data broadcasting schedules. As special cases, we consider a centralized and a decentralized broadcasting schedule. We determine the steady-state distribution of this model and derive closed-form expressions for performance metrics such as the retrieval time of a reliable estimate, the load of the system, the busy
period of the system.

The content of Chapter 2 is based on the following papers.


In Chapter 3, we consider the problem of retrieving a reliable estimate of an attribute monitored by a wireless sensor network within a fixed time window and with minimum energy consumption for the sensors. The sensors are assumed to be placed in the plane according to a random point process and to follow a sleep/awake cycle. We derive a closed-form expression for the expected energy consumption of sensors under a Greedy sensor transmission schedule, which schedules every time slot the closest possible sensor. To meet general time constraints, we formulate a stochastic dynamic programming model to determine a sensor transmission schedule such that a reliable estimate is retrieved with minimum energy consumption. We also develop a heuristic and an off-line, optimal schedule, which assumes that the sleep/awake state of the sensors is known ahead of time. We compare numerically the energy consumption of the schedule achieved using stochastic dynamic programming with the heuristic and the off-line, optimal schedule. The content of Chapter 3 is based on the following papers.


In Chapter 4, the data retrieval time of a reliable attribute monitored by a wireless sensor network is analyzed. The sensors are assumed to harvest energy from the environment in order to charge their batteries. Multiple clients arrive at the sensor network according to some random process and are interested in retrieving a reliable estimate of an attribute. Sensor broadcasting
is employed. However, broadcasting is influenced by the energy availability at the sensor, which depends on the energy harvesting process. We determine the distribution of the time for an arbitrary client to retrieve a reliable estimate from the sensor network when the sensors broadcast data and harvest energy in a decentralized fashion. We also derive the expected time for a client to retrieve a reliable estimate. Lastly, we determine closed-form expressions for the retrieval time of a reliable estimate when the sensors are equipped with asymptotically large batteries and energy harvesting rates. The content of Chapter 4 is based on the following working paper.


In Chapter 5 the Pareto front of a deployment cost of wireless caches in the plane and a retrieval cost of data from these caches is determined. Partitioning and coding storage strategies are considered. We provide an exact characterization of the Pareto fronts of the deployment and data retrieval costs under coding and partitioning strategies. We quantify the improvements offered by the optimal coding strategy in comparison to the partitioning strategy. Finally, we show that no coded deployment can be dominated by a partitioning strategy. The content of Chapter 5 is based on the following papers.


In Chapter 6 we analyze the situation when clients are interested in retrieving data about an area monitored by a wireless sensor network by means of queries. The interest is in responding to these queries in a timely manner and without compromising the quality (age) of the data. We trade-off between query response time and age of the data by dynamically assigning queries either to the sensor network, where the age of the data is negligible, but query response time may be large, since queries need to wait to be processed, or to a central database, where query response time is negligible since data was stored at a previous time, but these stored data is possibly outdated (has a large age). We use non-standard exponential uniformization
of a continuous-time Markov process with a drift to determine an optimal query assignment strategy. We provide a numerical comparative analysis of the performance of the optimal policy and of several heuristics, commonly used in practice. The results provide a formal support for the design and implementation of query assignment policies in practice so that the system can perform close to the optimum. The content of Chapter 6 is based on the following paper and book chapter.


CHAPTER 2

Data Retrieval Time in Wireless Sensor Networks

2.1 Introduction

In this chapter we consider the problem of retrieving a reliable estimate of an attribute, for instance, temperature, position of an object, which is monitored by a wireless sensor network. We assume that measurements from several, distinct sensors need to be acquired in order to compute a reliable estimate of this attribute. This can be the case, for instance, of detection applications where at least three sensor measurements are needed to compute the coordinates of an object. This can also be the case of sensors for which the accuracy of the measurements is influenced by the environment in which the sensors are deployed, by the position relative to the phenomenon/object monitored, and, thus, several sensor measurements are needed to acquire reliable information about the monitored phenomenon/object.

We focus on the analysis of the time needed for a reliable estimate of an attribute to be retrieved. Since measurements are acquired from several sensors, the mechanism according to which the sensors transmit their measurements influences the retrieval time of this reliable estimate. In this chapter, we analyze various sensor transmission schedules and the corresponding time to retrieve a reliable estimate from the sensors.

The model considered in this chapter is as follows. Consider a network of wireless sensors that have noisy measurements of an attribute. Clients, arriving at the network, are interested in collecting from the network a sufficiently large set of measurements from distinct sensors. Based on this set, they compute a reliable estimate of this attribute. The size of the set of measurements must be large enough so that the accuracy of the estimate is above a targeted threshold. The clients wait at the network until they collect a sufficient number of measurements to retrieve a reliable estimate, then leave the system.
Data estimation in wireless sensor networks has been studied extensively in [16, 86, 100–102]. Data estimation techniques combine data from several sensors at a fusion center to improve data accuracy, which is difficult to achieve when the measurement of a single sensor alone is collected. Such techniques are specifically used for networks consisting of small, inexpensive sensors that have a low sensing precision. Generally, an estimate is determined by performing a linear combination of the sensor measurements, a technique referred to as the centralized BLUE [86]. In [100, 101], the authors study the problem of energy minimization while keeping the mean square estimation error of the sensor measurements below a targeted threshold. In [102], the estimation of a noise-corrupted parameter under bandwidth constraints is considered. In [16], the problem of distributed data estimation with fading channels is considered. The variance of the estimate under perfect and partial channel information availability is considered.

Complementary to the work mentioned above, in this chapter we analyze the performance of sensor transmission schedules which support data estimation. Scheduling techniques for wireless sensor networks have been extensively studied in, for example, [14, 28]. However, little work has been done on sensor transmission schedules that support data gathering. In [93] round-robin and data generation rate-based schedules that support data gathering in sensor networks are considered. It is shown that in terms of the latency, the schedules considered outperform each other depending on the network topology.

In this chapter we focus on sensor scheduling techniques that support the retrieval of a reliable estimate of an attribute monitored by a wireless sensor network. In Section 2.1.1 we consider a basic model for the reliability of an estimate.

In Part I of this chapter we consider a time-slotted model where sensors transmit their measurements to a single client according to centralized and decentralized schedules. Moreover, the sensors follow an asleep/awake cycle. The main focus of this section is to compare the time to retrieve a reliable estimate under the decentralized schedule with the retrieval time under an optimal, centralized schedule. Our main contribution is to show that despite the contrasting settings of the two transmission schedules, the decentralized schedule provides a constant factor approximation to the optimal centralized schedule with respect to the retrieval time of a reliable estimate.

In Part II of this chapter we consider a continuous-time model where multiple clients are interested in retrieving a reliable estimate from the network. We assume that the sensors are broadcasting their measurements.
The main focus of this section is to provide a framework to determine the retrieval time of a reliable estimate under various broadcasting schedules. Our main contribution is to determine closed-form expressions for the steady-state distribution of the system under various broadcasting schedules and, consequently, to compute exactly metrics such as the retrieval time of a reliable estimate, the load of the system, the length of the busy period, i.e., the time at least one client is present at the sensor network.

2.1.1 Data Reliability Model

In this section we consider the following model for the reliability of an estimate of an attribute monitored by a wireless sensor network, which will be used throughout this chapter.

Consider a wireless sensor network consisting of $N$ sensors, where each sensor has a measurement of an attribute $\theta$. The measurements are subject to additive normally distributed noise with variance $\sigma^2$, i.e.,

$$X_i \sim N(\theta, \sigma^2).$$

The noise is independent and identically distributed across the sensors.

Clients arriving at the sensor network are interested in obtaining a reliable estimate $\bar{X}$ of $\theta$ based on the sensor measurements. We assume that an estimate is reliable if the variance of $\bar{X}$ is below a threshold $H$. The estimate can be obtained by retrieving an arbitrary set of $s$ sensor measurements such that:

$$\text{Var}(\bar{X}) = \text{Var}(\frac{1}{s} \sum_{i=1}^{s} X_i) = \frac{1}{s^2} \sum_{i=1}^{s} \text{Var}(X_i) = \frac{\sigma^2}{s}.$$

We consider $s = \lceil \frac{\sigma^2}{H} \rceil$. Since there are $N$ sensors that can provide at most $N$ measurements, we also assume that $s \leq N$.

2.2 Part I: Data Retrieval Time - Asleep/Awake Sensor Cycles

In this section, we consider the problem of retrieving a reliable estimate of an attribute monitored by a sensor network when the sensors follow an asleep/awake cycle. In particular, we analyze the retrieval time of a reliable estimate under centralized and decentralized transmission schedules.
2.2.1 Model and Problem Statement

We consider a slotted-time model with a single client interested in retrieving a reliable estimate $\bar{X}$ of an attribute $\theta$ from a wireless sensor network. The wireless sensors transmit their measurements to this client according to some transmission schedule. A time slot is defined as the time it takes for a sensor to transmit a measurement to the client.

We assume that simultaneous sensor transmissions lead to a destructive collision and the client does not obtain any information.

The sensors follow an asleep/awake cycle. More precisely, each sensor is awake every time slot with probability $p$, $0 < p < 1$, and asleep with probability $1 - p$. The probability of being awake is independent over time and across the sensors. Being awake or asleep reflects, for instance, the availability of energy in case of sensors that harvest energy from the environment. Moreover, we assume that sensors can only transmit if they are awake.

We consider a decentralized ($R$) and a centralized ($C$) sensor transmission schedule. Under the decentralized $R$ transmission schedule, every time slot, an awake sensor transmits its measurement with probability $q$, $0 < q < 1$, and remains silent with probability $1 - q$. In the analysis of this transmission schedule we will optimize over $q$. The $R$ schedule is similar to slotted ALOHA studied in [99]. Compared to [99], however, in this model, collisions do not provide any useful information to the receiver.

The optimal centralized $C$ schedule assumes that the sensors are centrally scheduled for transmission based on their asleep/awake status and whether the client has already received their measurement. More precisely, a sensor that has not successfully transmitted its measurement in the previous time slots is considered eligible. If two or more sensors are awake and eligible for transmission during the same time slot, then one of these sensors is randomly selected for transmission.

We are interested in the expected retrieval time $E[W_s]$, under the decentralized and centralized schedules, during which the client retrieves a reliable estimate $\bar{X}$ of attribute $\theta$.

We will use the digamma function, defined as $\psi(n) = H_{n-1} - \gamma$, $n \in \mathbb{N}$, where $H_n = \sum_{k=1}^{n} \frac{1}{n}$ is the $n^{th}$ harmonic number, and $\gamma$ is the Euler-Mascheroni constant.
2.2 Part I: Data Retrieval Time - Asleep/Awake Sensor Cycles

2.2.2 Analysis of Decentralized and Centralized Schedules

i) The Decentralized Schedule

In this section, we analyze the expected retrieval time for a client to retrieve $s$ measurements from distinct sensors, when the sensors transmit their measurements according to the decentralized $R$ schedule.

**Theorem 2.1.** For an optimal choice of $q$, the expected retrieval time for a client to retrieve $s$ distinct measurements under the decentralized $R$ schedule is:

$$
E[W^R_s] = \begin{cases} 
  f(N, s) \frac{1}{p(1-p)^{N-1}}, & \text{if } p \in (0, \frac{1}{N}) \\
  Nf(N, s)\left(\frac{N}{N-1}\right)^{N-1}, & \text{if } p \in \left[\frac{1}{N}, 1\right)
\end{cases}
$$

where $f(N, s) = [\psi(N + 1) - \psi(N - s + 1)]$.

**Proof.** Since the scheduling mechanism is independent and identical over time, the time it takes for the client to obtain the $i$-th distinct measurement after having received the $(i - 1)$-th measurement is the time until the first success in a Bernoulli trail.

The $i$-th measurement is successfully received if one of the $N - i + 1$ sensors from which a measurement has not yet been retrieved, is the single sensor transmitting, i.e. the probability of success is

$$w_i = \binom{N}{1}pq(1 - pq)^{N-1} \frac{N - i + 1}{N}.
$$

Now, the retrieval time under the $R$ schedule is:

$$
E[W^R_s] = \sum_{i=1}^{s} \frac{1}{w_i} = \sum_{i=1}^{s} \frac{1}{pq(1 - pq)^{N-1} (N - i + 1)} = \psi(N + 1) - \psi(N - s + 1).
$$

(2.2)

It remains to optimize over $q \in [0, 1]$. The gradient of $E[W^R_s]$ is

$$
\frac{d}{dq} E[W^R_s] = \frac{\psi(N + 1) - \psi(N - s + 1)}{p} \frac{Npq - 1}{q^2(1 - pq)^N}.
$$

This shows that if $p \geq \frac{1}{N}$, then $E[W^R_s]$ is minimized for $q = \frac{1}{Np}$. In the case that $p < \frac{1}{N}$, the value of $q$ for which $E[W^R_s]$ is minimized is $q = 1$. \qed
Figure 2.1 shows that expected time to retrieve $s = 4$ measurements from distinct sensor, when the decentralized transmission schedule $R$ is employed. For a sufficiently large size of the sampling sensor set $N$, $E[W^R_s]$ is independent of the transmission probability $p$ and becomes a function of $N$. This is explained by the fact that the probabilities of one sensor transmitting and being awake compensate each other in order to minimize the retrieval time of a reliable estimate $\overline{X}$. For example, a low probability of being awake is compensated by a high probability of transmitting when awake such that the retrieval time is minimized.

Figure 2.1: The expected time $E[W^R_s]$ to retrieve $s = 4$ measurements from distinct sensors need to be retrieved, for different transmission probabilities, under the decentralized schedule $R$.

ii) The Centralized Schedule

In this section, we analyze the expected retrieval time for a client to retrieve a reliable estimate $\overline{X}$, i.e., to retrieve $s$ measurements from distinct sensors, when the sensors transmit their measurements according to the centralized $C$ schedule.

**Theorem 2.2.** The expected retrieval time for the client to retrieve $s$ distinct mea-
The expected retrieval time under the centralized $C$ schedule is:

$$\mathbb{E}[W^C_s] = \sum_{i=1}^{s} \frac{1}{1 - (1 - p)^{N-i+1}}.$$  

Proof. Again, the scheduling mechanism is independent and identical over time. Therefore, the time until the client retrieves the $i$-th sensor measurement, given it already received $i - 1$ distinct measurements, can be viewed as the time until a first success in a Bernoulli trial, where the success probability $v_i$ is the probability that one sensor of those that have not previously transmitted is awake. Hence, the probability of success $v_i$ is:

$$v_i = 1 - (1 - p)^{N-i+1}$$

Now, the expected retrieval time under the $C$ schedule is:

$$\mathbb{E}[W^C_s] = \sum_{i=1}^{s} \frac{1}{v_i} = \sum_{i=1}^{s} \frac{1}{1 - (1 - p)^{N-i+1}}.$$  

\[2.2.3 \text{ Decentralized vs. Centralized Transmission Schedule}\]

In this section we compare the retrieval time of an estimate under the decentralized and centralized schedules for large-size sensor networks.

**Theorem 2.3.** Let $s \in \mathbb{N}$. Then

$$\lim_{N \to \infty} \frac{\mathbb{E}[W^R_s]}{\mathbb{E}[W^C_s]} = e.$$  

Proof. For $N$ sufficiently large,

$$\lim_{N \to \infty} \mathbb{E}[W^R_s] = \lim_{N \to \infty} \sum_{i=1}^{s} \frac{N}{N - i + 1} \left(1 + \frac{1}{N-1}\right)^{N-1}$$

$$= se.$$  

Also, considering $p$ to be fixed, from Theorem 2.2, it follows that

$$\lim_{N \to \infty} \mathbb{E}[W^C_s] = s$$

Using (2.3) and (2.4), the result follows.
Figure 2.2 shows the difference in the retrieval time of \( s = 10 \) measurements from distinct sensors under the decentralized \( R \) and centralized \( C \) transmission schedule and for various sizes, \( N \), of the sensor network. In the case of large sensor networks, the ratio of the expected retrieval time under the decentralized and centralized schedules approaches a constant \( e \) from above. Also, for sufficiently large \( N \), the difference in the retrieval time between the two schedules is bounded from above by a constant.

We now consider \( s \) to be a constant fraction smaller than the sampling sensor size \( N \), with \( N/s = \beta, \beta > 1 \).

**Theorem 2.4.** For sufficiently large \( N \) and \( s \) and a constant ratio \( \beta = N/s \), where \( \beta > 1 \),

\[
\frac{\mathbb{E}[W^R_s]}{\mathbb{E}[W^C_s]} \leq e \lceil \beta \rceil \log \left( \frac{\beta}{\beta - 1} \right).
\]

**Proof.** We now consider both \( N \) and \( s \) sufficiently large and keep the ratio \( \beta = N/s \) constant, where \( \beta > 1 \).

\[
\mathbb{E}[W^R_s] = \left\lfloor \beta s \right\rfloor \sum_{x=\left\lfloor \beta s \right\rfloor - s + 1}^{\left\lfloor \beta s \right\rfloor} \frac{1}{x} \cdot \left( \frac{\left\lfloor \beta s \right\rfloor}{\left\lfloor \beta s \right\rfloor - 1} \right)^{\left\lfloor \beta s \right\rfloor - 1}
\]

\[
\leq \left\lfloor \beta s \right\rfloor \sum_{x=\left\lfloor \beta s \right\rfloor - s + 1}^{\left\lfloor \beta s \right\rfloor} \frac{1}{x} \cdot \left( \frac{\left\lfloor \beta s \right\rfloor}{\left\lfloor \beta s \right\rfloor - 1} \right)^{\left\lfloor \beta s \right\rfloor - 1}
\]

\[
\leq \left\lfloor \beta s \right\rfloor \int_{\beta s - s + 1}^{\beta s} \frac{1}{x - 1} dx \cdot \left( \frac{\left\lfloor \beta s \right\rfloor}{\left\lfloor \beta s \right\rfloor - 1} \right)^{\left\lfloor \beta s \right\rfloor - 1}
\]

\[
\leq \left\lfloor \beta \right\rfloor s \log \left( \frac{\beta s}{\beta s - s} \right) \left( \frac{\left\lfloor \beta s \right\rfloor}{\left\lfloor \beta s \right\rfloor - 1} \right)^{\left\lfloor \beta s \right\rfloor - 1} \tag{2.5}
\]

Using (2.5) and the fact that \( \mathbb{E}[W^C] \geq s \), see (2.4),

\[
\frac{\mathbb{E}[W^R_s]}{\mathbb{E}[W^C_s]} \leq \left\lfloor \beta \right\rfloor \log \left( \frac{\beta}{\beta - 1} \right) \cdot \left( 1 + \frac{1}{\left(\left\lfloor \beta s \right\rfloor - 1 \right)^{\left\lfloor \beta s \right\rfloor - 1}} \right)
\]

Note that \( \left( 1 + \frac{1}{\left(\left\lfloor \beta s \right\rfloor - 1 \right)^{\left\lfloor \beta s \right\rfloor - 1}} \right)^{\left\lfloor \beta s \right\rfloor - 1} \) approaches \( e \) from below. Therefore,

\[
\frac{\mathbb{E}[W^R_s]}{\mathbb{E}[W^C_s]} \leq e \left\lfloor \beta \right\rfloor \log \left( \frac{\beta}{\beta - 1} \right).
\]
Also note that \( \lim_{N \to \infty} \lceil \beta \rceil \log(\frac{\beta}{\beta - 1}) = 1 \), in accordance with Theorem 2.3.

Figure 2.3 shows that for a fixed ratio \( \beta \) of the size of the sampling sensor set \( N \) and the \( s \) retrieved measurements, where \( N \) and \( s \) are large, the retrieval time under the decentralized schedule is no higher than a constant \( e \lceil \beta \rceil \log(\frac{\beta}{\beta - 1}) \) than the centralized schedule.

### 2.3 Part II: Data Retrieval Time under Wireless Sensor Broadcasting

In contrast with the previous model, in this section, we consider a continuous-time model where multiple clients are interested in retrieving a reliable estimate of an attribute monitored by a wireless sensor network and sensors broadcast their measurements. We also assume that sensors are always able to broadcast their measurements, whereas in the previous model transmissions were hindered by the fact that sensor followed an asleep/awake cycle and, thus, sensors were able to transmit only if they were awake. Lastly, in this section we consider a setting where sensors do not interfere with each other, whereas in the previous model simultaneous sensor transmissions led to a destructive collision and all information involved in this collision was assumed lost.

In this section we analyze the retrieval time of a reliable estimate, the load of the system and the length of the busy period, i.e, the time at least one client is present at the network, under various sensor broadcasting schedules. In Section 2.3.4 we propose a simple model to analyze the performance of a generic broadcasting schedule. This model is further extended in Sections 2.3.6 and 2.3.9, where we analyze the performance of a decentralized and a centralized broadcasting schedule.

#### 2.3.1 Model and Problem Statement

Consider a model where clients, arriving at the sensor network according to a Poisson process with rate \( \lambda_a \), are interested in retrieving from the network a reliable estimate \( \bar{X} \) of attribute \( \theta \).

At an exponential rate \( \mu \) a sensor broadcasts its measurement. The broadcasting sensor is specified by some broadcasting schedule. This broadcasted measurement is collected by a client only if it is the first time this sensor broadcasts its measurement to this client. We call such a measurement to
Figure 2.2: The expected time to retrieve $s = 10$ measurements from distinct sensors under the decentralized $R$ and the centralized $C$ sensor transmission schedule and for various sizes, $N$, of the sensor network.

Figure 2.3: The limiting ratio $E[W_s^R]/E[W_s^C]$ for large $N$, when $\beta = 2$. 
be *innovative* for this client. If, for instance, a sensor broadcasts its measurement twice within the time that a particular client is in the system, then the second transmission will be redundant for that client.

As special cases, we analyze centralized and decentralized broadcasting. Under centralized broadcasting, the sensors are centrally scheduled for broadcasting such that the retrieval time of the client waiting the longest at the network is minimum. Moreover, the centralized broadcasting assumes that the set of measurements collected by the client waiting the longest at the network is known. Under the decentralized broadcasting, the sensors broadcast independently, at random. In addition, the set of measurements collected by the clients is not known.

The problem of providing a fixed number of units of service (in our case, a set of measurements) to independently and identically distributed arriving clients has been studied in, for instance, [23, 41, 64]. In [23] a discrete-time multi-server queueing model is considered where each client is interested in receiving \( s \) units of service. Every time slot, \( m \) units of service are provided at random to \( m \) clients. In [64] a discrete time queueing system with a First-Come-First-Serve service discipline is assumed. In contrast to the work mentioned above, we model a multi-class queueing system by means of a continuous-time model, where clients must obtain a sufficiently large set of observations from distinct sensors.

We are interested in the expected time for clients to retrieve a reliable estimate, \( \mathbb{E}[W_s] \), the expected number of clients in the system, \( \mathbb{E}[L_s] \), and the expected length of the busy period of the system, \( \mathbb{E}[K_s] \).

### 2.3.2 Preliminaries

In this section we show that the model in Section 2.3.1 can be formulated as a multi-class queue with the following properties:

- There are \( s \) classes of clients, labeled \( 0, \ldots, s - 1 \). Class \( i \) clients are those clients that have collected \( i \) measurements.

- Class 0 clients enter the queue according to a Poisson process with rate \( \lambda_a \).

- If a class \( i \) client receives service, then all other clients of that class, as well as all class \( 0, 1, \ldots, i \) clients receive service simultaneously at that moment.

- Class \( i < s - 1 \) clients that receive service re-enter the queue as clients of class \( i + 1 \). Class \( s - 1 \) clients that receive service leave the system.
In order to show the above properties, we start by considering a continuous-time Markov chain on a state space $\mathbb{N}_0^s$, where $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$. Let the state $n \in \mathbb{N}_0^s$ be a vector $n = (n_0, n_1, \ldots, n_{s-1})$ where $n_i \geq 0$, $0 \leq i \leq s - 1$, denotes the case where each client has to collect $s$ measurements from distinct sensors and there are $n_i$ clients which have collected $i$ measurements from distinct sensors. We denote by class $i$, $0 \leq i \leq s - 1$, all the clients that have retrieved $i$ sensor measurements.

As an example, for $s = 4$, the state $n = (1, 3, 0, 2)$ corresponds to the case where there is one class 0 client with no measurements, three class 1 clients, each having one measurement, no class 2 clients and, lastly, two class 3 clients, each having 3 measurements (see also Figure 2.4).

Finally, let $Q_i$ denote the $i$–th client arriving at the system, with $i \geq 1$.

\begin{figure}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
$\lambda_n$ & $Q_6:|$ & $Q_3:[X_2]|$ & $Q_1:[X_2][X_4][X_1]$ \\
Class 0 & ($n_0 = 1$) & Class 1 & Class 2 & Class 3 \\
& ($n_1 = 3$) & & ($n_2 = 0$) & ($n_3 = 2$) \\
\hline
\end{tabular}
\caption{Example with 6 clients in the system, $s = 4$. Clients $Q_1$ and $Q_2$ have 3 measurements $\{X_1, X_2, X_4\}$ and need one more measurement to be able to compute a reliable estimate. As soon as these clients retrieve a measurement from a sensor distinct from sensors 1, 2, 4, they leave the system. There are no clients that have 2 measurements. There are 3 clients, $Q_3, Q_4, Q_5$, that have retrieved one measurement $\{X_2\}$. As soon as a sensor, which is distinct from sensor 2, broadcasts, these three clients collect a second measurement and become of class 2. Client $Q_6$ has no measurements. As soon as a broadcasts occurs, from an arbitrary sensor, this client becomes of class 1.}
\end{figure}

2.3.3 Behaviour of the System under Broadcasting

We state the following claims about the behavior of the classes of clients under broadcasting, thereby establishing the properties stated above. Figure 2.5 shows the behavior of the system over time, when arbitrary sensors broadcast their measurements to the clients.

**Claim 2.1.** All class $i$ clients, $1 \leq i \leq s - 1$, have the same set of measurements, which we denote by $C_i$. 
2.3 Part II: Data Retrieval Time under Wireless Sensor Broadcasting

Measurements to collect

![Diagram of sensor broadcasting and measurement retrieval over time](image)

Figure 2.5: Example of sensor broadcasting and measurement retrieval over time, \( s = 3 \) (each client needs to retrieve 3 measurements from distinct sensors). Clients \( Q_i, 1 \leq i \leq 3 \), arrive according to some Poisson process. Observations \( X_i \) are broadcasted over time from arbitrary sensors. In this example, \( Q_1 \) collects first \( X_1 \) and \( X_2 \), then drops \( X_1 \) two consecutive times, then collects \( X_4 \) and leaves the system as it has collected sufficient measurements to compute a reliable estimate. Both \( Q_2 \) and \( Q_3 \) collect \( X_1 \), \( X_4 \) and \( X_2 \), then leave the system at the same time.

**Proof.** Without loss of generality, assume that we start with an empty system. All clients that arrive until the first broadcast are of class 0. Upon a broadcast \( \{X_k\} \), all these clients change class from 0 to 1 and, clearly, collect the same measurement \( X_k \). Between the first and the second broadcast, all new clients arriving at the network are of class 0. Upon a second broadcast \( X_j \), two cases can occur: 1) the second broadcast \( X_j \) is innovative for class 1 clients and also for class 0, i.e. \( k \neq j \). Then class 1 clients change class from 1 to 2 and all these clients have the same two measurements \( \{X_k, X_j\} \). Also class 0 clients change class from 0 to 1 and they all have the same measurement \( \{X_j\} \); 2) the second measurement \( X_j \) is not innovative for class 1 clients, but is innovative for class 0 clients, i.e. \( k = j \) (the broadcast is from the same sensor that transmitted the first time). Then only class 0 clients change class from 0 to 1 and collect \( X_k \). Thus, upon the second transmission, all class 1 clients have the same measurement \( \{X_k\} \). By this inductive reasoning, all class \( k \) clients, \( 0 \leq k \leq s - 1 \), have the same set of measurements. \( \square \)

**Claim 2.2.** If a measurement is innovative for class \( k \) clients, \( 1 \leq k \leq s - 1 \), then this observation is innovative for all class \( j \) clients, where \( 0 \leq j \leq k \).

**Proof.** We prove this by contradiction. First, by Claim 2.1 all clients of a class have the same set of measurements. Suppose that a measurement is innova-
tive for class $k$ clients, $1 \leq k \leq s - 1$, but is not innovative for class $j$ clients, $j < k$. This means that class $j$ clients have previously collected this measurement. Also, class $j$ clients have arrived after class $i$ clients (otherwise, they would have been at least class $k$). Thus, when class $j$ clients collected the measurement, class $k$ clients were present in the system, but they did not collect it even though they did not have the measurement. This contradicts the initial assumption that clients always collect innovative measurements. This concludes the proof.

**Claim 2.3.** Class $j$ clients have a subset of the set of measurements of class $i$ clients, $0 \leq j < i$.

**Proof.** From Claims 2.1 and 2.2.

Lastly, we make the following simple observation.

**Claim 2.4.** Under broadcasting, the system is stable for any $\mu > 0$.

The system is stable due to the fact that at a positive rate, which depends on the broadcasting schedule and is at most $\mu$, all clients of a class $i$, $1 \leq i \leq s - 1$, will receive an innovative measurement and will change class from $i$ to $i + 1$ (the rate at which clients change class is $\mu$ if upon each broadcast the measurement is innovative for that class). The time a client waits within a class is, thus, finite and the number of classes, $s$, is also finite. As a result, for any $\mu > 0$, the retrieval time of an estimate for a client is finite.

In the next section we will formulate the properties of the multi-class queue more precisely. In particular, we will describe it as a continuous-time Markov chain and specify all transition probabilities. Moreover, we will provide the stationary distribution of this chain.

### 2.3.4 Continuous-Time Markov Chain: General Multi-class Queue for Broadcasting

We consider the continuous-time Markov chain corresponding to an arbitrary sensor scheduling mechanism. In this section, we will not consider any particular details of this schedule. Instead, we will start from a given set of service rates for each of the classes in the queue. In the next two sections, we will then show which service rates correspond to various specific sensor scheduling mechanisms.

We first introduce the following notation. Let $e_0$ be a vector of length $s$ with zero entries, except the first entry, which has value 1. Let $U_i : \mathbb{N}_0^s \rightarrow \mathbb{N}_0^s$, where $\mathbb{N}_0^s$ is the set of non-negative integers with length $s$.

0 ≤ i ≤ s − 1, be defined as $U_i(n) = (0, n_0, \ldots, n_{i-1}, n_i + n_{i+1}, n_{i+2}, \ldots, n_{s-1})$, i.e. clients change class from $j$ to $j + 1$, 0 ≤ $j$ ≤ $i$. As an example, for $s = 4$ and $n = (2, 1, 3, 5)$, $U_2(n) = (0, 2, 1, 8)$. See also Figure 2.6.

We have the following basic structure for the chain. Let us first consider $n$ with all non-zero entries. Then, for each 0 ≤ $i$ ≤ $s − 1$, a transition occurs from $n$ to $U_i(n)$ at a rate $\mu_i$. Also, there is a transition from $n$ to $n + e_0$ at an exponential rate $\lambda_a$.

If $n$ contains zero entries, some care is required with this basic structure. We illustrate this by means of two examples.

Firstly, let 0 < $i$ ≤ $s − 1$ and consider $n_j = 0$ if and only if $j = i$. In this case $U_i(n) = U_{i-1}(n)$. Also, the transition rate from $n$ to $U_{i-1}(n)$ is $\mu_{i-1} + \mu_i$. As an example, consider $n = (2, 0, 3, 5)$, i.e. $i = 1$ since $n_1 = 0$. Then $U_1(n) = U_0(n) = (0, 2, 3, 5)$. Thus, at rate $\mu_3$ a transition occurs from $n$ to $(0, 2, 0, 3)$, at rate $\mu_2$ a transition occurs from $n$ to $(0, 2, 0, 8)$ and at rate $\mu_1 + \mu_0$ a transition occurs from $n$ to $(0, 2, 3, 5)$.

Secondly, let 0 < $i$ ≤ $s − 1$ and consider $n_j = 0$ if and only if $j ≤ i$. In this case $U_0(n) = U_1(n) = \cdots = U_i(n) = n$. As an example, consider $n = (0, 0, 0, 5)$, i.e. $i = 2$ since $n_0 = n_1 = n_2 = 0$. Then $U_0(n) = U_1(n) = U_2(n) = n = (0, 0, 0, 5)$. A transition occurs from $n$ to $(0, 0, 0, 0)$ at rate $\mu_3$. Observe that, in this case, the rate at which the classes with non-zero clients change class is not affected by the classes with zero clients. We will refer to the classes with zero clients which are smaller than the smallest class with at least one client as ‘leading zeros’ in $n$.

For notational convenience, for 1 ≤ $i$ ≤ $s$ and 0 ≤ $x$ ≤ $n_i$, consider $L_{i-1,x} : \{0\} \times \mathbb{N}_0^{s-1} \rightarrow \mathbb{N}_0^s$ to be defined as $L_{i-1,x}(n) = (n_1, \ldots, n_{i-1}, x, n_i - x, n_{i+1}, \ldots, n_{s-1})$. Finally, for $x ≥ 0$, let $L_{s-1,x}(n) = (n_1, \ldots, n_{s-1}, x)$. As an example, for $n = 4$, $L_{1,x}((0, 1, 3, 5)) = (1, x, 3 - x, 5)$ and $L_{3,x}((0, 1, 3, 5)) = (1, 3, 5, x)$. See also Figure 2.6. Note that the reason for introducing $L_{i,x}$ is that, under the condition that $n_0 = 0$, $U_i(L_{i,x}(n)) = n$ and thus, $L_{i,x}(n)$
provides a convenient means to describe the states for which there is a transitions into \( n \). As an example, consider \( n = (0, 2, 3, 5) \). Then \( L_{1,x}(n) = (2, x, 3 - x, 5) \) and \( U_1(L_{1,x}(n)) = U_1(2, x, 3 - x, 5) = n \).

**Steady-State of the System**

In this section we formulate the balance equations of system described above. Let \( \Upsilon(n) \) denote the long-run probability of the system being in state \( n \), i.e. the long-run probability of having \( n_i, 0 \leq i \leq s - 1 \), client of class \( i \) present in the system.

**Lemma 2.1.** If \( n_0 > 0 \) then

\[
\Upsilon(n) \left( \lambda_a + \sum_{i=0}^{s-1} \mu_i \right) = \lambda_a \Upsilon(n - e_0), \tag{2.6}
\]

else

\[
\Upsilon(n) \left( \lambda_a + \sum_{i=0}^{s-1} \mu_i \right) = \sum_{i=0}^{s-2} \sum_{n_i+1} n_{i+1} \Upsilon(L_{i,x}(n)) + \sum_{x=0}^{\infty} \mu_{s-1} \Upsilon(L_{s-1,x}(n)). \tag{2.7}
\]

**Proof.** The balance equations follow from the analysis of the transitions into and out of state \( n \). Equation (2.7) has been obtained by adding self transitions for the classes \( 1 \leq j \leq i \), where \( i = \min\{k \mid n_k > 0, 0 \leq k \leq s - 1\} \), i.e. for the ‘leading zeros’ in \( n \).

We next determine the steady-state distribution of this chain. For ease of notation, let \( \bar{\mu}_i = \sum_{k=i}^{s-1} \mu_k \). This is the total rate at which class \( i \) changes class to \( i + 1 \). This because class \( i \) changes to class \( i + 1 \) every time class \( j \geq i \) changes class to \( j + 1 \) (see also Figure 2.6). Note that \( \bar{\mu}_0 = \sum_{k=0}^{s-1} \mu_k \) and \( \bar{\mu}_{s-1} = \mu_{s-1} \).

**Theorem 2.5.** The steady-state distribution of the continuous-time Markov chain defined above is:

\[
\Upsilon(n) = \prod_{i=0}^{s-1} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_i} \right)^{n_i} \prod_{i=0}^{s-1} \frac{\bar{\mu}_i}{\lambda_a + \bar{\mu}_i}, \tag{2.8}
\]

where \( n_i \geq 0 \) for \( 0 \leq i \leq s - 1 \).
Proof. First, it is easy to verify that (2.8) satisfies (2.6) for \( n_0 > 0 \).

For \( n_0 = 0 \), we use induction on \( s \) to prove that (2.8) satisfies (2.7). As a base case, we consider \( s = 1 \). For \( s = 1 \), and observing that in this case \( \bar{\mu}_0 = \mu_0 \), (2.7) reduces to

\[
\Upsilon(0)(\lambda_a + \mu_0) = \mu_0 \sum_{x=0}^{\infty} \Upsilon(L_0,x(n)) = \mu_0 \sum_{x=0}^{\infty} \Upsilon(x).
\]  

(2.9)

Using (2.8) in (2.9), we have that

\[
\frac{\mu_0}{\lambda_a + \mu_0} (\lambda_a + \mu_0) = \mu_0 \sum_{x=0}^{\infty} \left( \frac{\lambda_a}{\lambda_a + \mu_0} \right)^x \frac{\mu_0}{\lambda_a + \mu_0},
\]  

(2.10)

which is trivially satisfied.

Before considering the general case \( s > 1 \), for clarity, observe that, using (2.8), for \( 0 \leq i \leq s - 1 \),

\[
\Upsilon(L_i,x(n)) = \Upsilon(n_1, \ldots, n_i, x, n_{i+1} - x, n_{i+2}, \ldots, n_{s-1})
\]

\[
= \prod_{j=0}^{i-1} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_j} \right)^{n_j+1} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_i} \right)^x \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_{i+1}} \right)^{n_{i+1} - x}
\]

\[
\cdot \prod_{j=i+2}^{s-1} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_j} \right)^{n_j} \prod_{i=0}^{s-1} \frac{\bar{\mu}_i}{\lambda_a + \bar{\mu}_i},
\]  

(2.11)

where we note that the exponent of the first \( i \) terms is \( n_{j+1} \).

We now consider the general case \( s > 1 \) for induction. First, we consider (2.7) for the \( s-1 \) dimensional system with service rates \( \mu_0, \mu_1, \ldots, \mu_{s-3}, \mu_{s-2} + \mu_{s-1} \). Note that \( \bar{\mu}_{s-2} = \mu_{s-2} + \mu_{s-1} \). We assume that (2.7) for this \( s-1 \) dimensional system has the solution provided in (2.8). Based on this induction hypothesis, we will show that for a \( s \) dimensional system (2.7) has the solution provided in (2.8).

Now, writing (2.7) for this \( s-1 \) dimensional system in detail with the
product-form stationary distribution according to (2.8) gives

\[
\prod_{i=0}^{s-2} \left( \frac{\lambda_i}{\lambda_i + \mu_i} \right)^{n_i} (\lambda_i + \bar{\mu}) = 
\sum_{i=0}^{s-3} \sum_{n=0}^{i+1} \mu_i \prod_{j=0}^{i-1} \left( \frac{\lambda_j}{\lambda_j + \mu_j} \right)^{n_{j+1}} \left( \frac{\lambda_j}{\lambda_j + \mu_j} \right)^x \prod_{j=i+2}^{s-2} \left( \frac{\lambda_j}{\lambda_j + \mu_j} \right)^{n_j}
+ \bar{\mu}_{s-2} \sum_{x=0}^{\infty} \prod_{i=0}^{s-3} \left( \frac{\lambda_i}{\lambda_i + \mu_i} \right)^{n_{i+1}} \left( \frac{\lambda_i}{\lambda_i + \mu_i} \right)^x \gamma_s (2.12)
\]

Note that in both sides of (2.12), the constant \( \prod_{i=0}^{s-2} \frac{\bar{\mu}_i}{\lambda_i + \mu_i} \) has been divided out (this constant depends only on the size of \( n \) and on the service rates).

We next show that (2.7) has the solution provided in (2.8) for an s-dimensional system. Let \( \gamma_s = \prod_{i=0}^{s-1} \frac{\bar{\mu}_i}{\lambda_i + \mu_i} \). Multiplying all terms in (2.12) with \( \frac{\lambda}{\lambda + \bar{\mu}_s} \) gives

\[
\Upsilon(n) (\lambda + \bar{\mu}) = \sum_{i=0}^{s-3} \sum_{n=0}^{i+1} \mu_i \Upsilon(L_{i,x}(n))
+ \bar{\mu}_{s-2} \sum_{x=0}^{\infty} \prod_{i=0}^{s-3} \left( \frac{\lambda_i}{\lambda_i + \mu_i} \right)^{n_{i+1}} \left( \frac{\lambda_i}{\lambda_i + \mu_i} \right)^x \gamma_s
= \sum_{i=0}^{s-3} \sum_{n=0}^{i+1} \mu_i \Upsilon(L_{i,x}(n))
+ (\lambda + \bar{\mu}_{s-2}) \prod_{i=0}^{s-3} \left( \frac{\lambda_i}{\lambda_i + \mu_i} \right)^{n_{i+1}} \left( \frac{\lambda_i}{\lambda_i + \mu_i} \right)^{n_{s-1}} \gamma_s, (2.13)
\]

where we have used the fact that \( \bar{\mu}_{s-2} \sum_{x=0}^{\infty} \left( \frac{\lambda_i}{\lambda_i + \mu_i} \right)^x = \lambda_i + \bar{\mu}_{s-2} \).

Note that \(|n| = s\) in (2.13). It remains to show that the second term of the right-hand side of (2.13) equals:

\[
(\lambda + \bar{\mu}_{s-2}) \prod_{i=0}^{s-3} \left( \frac{\lambda_i}{\lambda_i + \mu_i} \right)^{n_{i+1}} \left( \frac{\lambda}{\lambda + \mu_{s-1}} \right)^{n_{s-1}} \gamma_s = \bar{\mu}_{s-2} \sum_{x=0}^{\infty} \Upsilon(L_{s-2,x}(n))
+ \mu_{s-1} \sum_{x=0}^{\infty} \Upsilon(L_{s-1,x}(n)). (2.14)
\]
The first term in the right-hand side of (2.14) is,

$$\mu_{s-2} \sum_{x=0}^{n_{s-1}} \Upsilon(L_{s-2,x}(n)) = \mu_{s-2} \prod_{i=0}^{s-3} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_{i}} \right)^{n_{i+1}} \cdot \sum_{x=0}^{n_{s-1}} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_{s-2}} \right)^x \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_{s-1}} \right)^{n_{s-1}-x} \gamma_s$$

$$= \prod_{i=0}^{s-3} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_i} \right)^{n_{i+1}} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_{s-1}} \right)^{n_{s-1}} \cdot \left( \lambda_a + \bar{\mu}_{s-2} - \frac{(\lambda_a + \bar{\mu}_{s-1})^{n_{s-1}+1}}{(\lambda_a + \bar{\mu}_{s-2})^{n_{s-1}}} \right) \gamma_s,$$

where in (2.15) we used that $\mu_{s-2} = \bar{\mu}_{s-2} - \bar{\mu}_{s-1}$ and that

$$\sum_{x=0}^{n_{s-1}} \left( \frac{\lambda_a + \bar{\mu}_{s-1}}{\lambda_a + \bar{\mu}_{s-2}} \right)^x \frac{\lambda_a + \bar{\mu}_{s-2} - (\lambda_a + \bar{\mu}_{s-1})(\lambda_a + \bar{\mu}_{s-1})^{n_{s-1}}}{\bar{\mu}_{s-2} - \bar{\mu}_{s-1}}.$$

The second term in the right-hand side of (2.14) is

$$\mu_{s-1} \sum_{x=0}^{\infty} \Upsilon(L_{s-1,x}(n)) = \mu_{s-1} \prod_{i=0}^{s-2} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_i} \right)^{n_{i+1}} \sum_{x=0}^{\infty} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_{s-1}} \right)^x \gamma_s$$

$$= \prod_{i=0}^{s-2} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_i} \right)^{n_{i+1}} (\lambda_a + \bar{\mu}_{s-1}) \gamma_s,$$

where in the last equation we used that $\bar{\mu}_{s-1} = \mu_{s-1}$ and that

$$\sum_{x=0}^{\infty} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_{s-1}} \right)^x \frac{\lambda_a + \bar{\mu}_{s-1}}{\bar{\mu}_{s-1}}.$$

Equation (2.14) now follows from (2.15) and (2.16) (for a detailed proof, see Section 2.5). The proof that (2.8) satisfies (2.7) follows from (2.13) and (2.14).

### 2.3.5 Performance Measures

In this section, we derive the following performance measures of the system.
Theorem 2.6. The expected number of clients in the system is:

\[ E[L_s] = \sum_{i=0}^{s-1} \frac{\lambda_a}{\mu_i}. \]

Proof. From Theorem 2.5.

Theorem 2.7. The expected retrieval time of \( s \) measurements from distinct sensors for a client is:

\[ E[W_s] = \sum_{i=0}^{s-1} \frac{1}{\mu_i}. \]

Proof. From Theorem 2.6 and Little’s Law.

Note that the time to retrieve any number \( k < s \) of sensor measurements can now directly be obtained from Theorem 2.5.

Theorem 2.8. The expected length of the busy period is:

\[ E[K_s] = \lambda^{-1} \left( \prod_{i=0}^{s-1} \frac{\bar{\mu}_i}{\lambda_a + \bar{\mu}_i} \right)^{-1} - 1. \]

Proof. Let \( p_e \) be the fraction of time the system is empty. From Theorem 2.5,

\[ p_e = \prod_{i=0}^{s-1} \frac{\bar{\mu}_i}{\lambda_a + \bar{\mu}_i}. \]

The result follows from interpreting the system as a renewal reward process. Then,

\[ p_e = \frac{1/\lambda_a}{1/\lambda_a + E[K_s]}, \]

where \( 1/\lambda_a \) is the expected period the system is empty.

So far we have considered a general broadcasting schedule for the sensors. In the next two sections we consider two specific broadcasting schedules. The analysis of these schedules is based on the model proposed in Section 2.3.4. In Section 2.3.6 we analyze a decentralized broadcasting schedule, under which sensors broadcast independently, at random. In Section 2.3.9 we analyze a centralized schedule which takes into account the residual time for a client to retrieve a reliable estimate from the sensors. We also show that the centralized broadcasting schedule considered is optimal with respect to the retrieval time of a reliable estimate.
2.3.6 Decentralized Broadcasting

Consider a decentralized broadcasting schedule, denoted by $D$, where sensors are scheduled for broadcasting independently, at random, i.e., at rate $\mu$, a sensor $i, 1 \leq i \leq N$, broadcasts with probability $1/N$. Thus, at rate $\mu/N$ one of the $N$ sensors broadcasts its measurement, independent of the other sensors. Figure 2.5 shows an example of a realization of the system under the $D$ broadcasting schedule.

2.3.7 Analysis of Decentralized Broadcasting Schedule $D$

Broadcasting schedule $D$ is characterized by the Markov chain introduced in Section 2.3.4 under the assumption that $\mu_i = \mu/N$ for $0 \leq i < s - 1$ and $\mu_{s-1} = \mu(1 - (s - 1)/N)$. This can be explained as follows. If all entries of $n$ are non-zero, then, for $0 \leq i < s - 1$, $\mu_i = \frac{\mu}{N}$ is the rate at which the broadcasted measurement belongs to $C_{i+1} \setminus C_i$. Recall that $|C_{i+1} \setminus C_i| = 1$ and this measurement is broadcasted at rate $\mu/N$, which is precisely the behavior of the $D$ schedule. Also, $\mu_{s-1} = \mu \left(1 - \frac{s-1}{N}\right)$ since class $s - 1$ has $s - 1$ measurements and an measurement is innovative for this class with probability $1 - (s - 1)/N$.

As an example, consider the state $n = (3, 1, 5, 7)$. Under the $D$ schedule, at rate $\mu/N$ a transition occurs from $n$ to $(0, 3, 6, 7)$ due to a measurement being innovative for class 1 clients, but not innovative for class 2 clients. A transition from $n$ to $(0, 3, 1, 5)$ occurs at rate $\mu(N - 3)/N$ due to a measurement being innovative for class 3 clients. Observe that when an measurement is innovative for class 3, then this measurement is innovative for all classes $i < 3$ (see Claim 2.2). Therefore, in this case, all classes change class from $i$ to $i + 1, 0 \leq i \leq 3$.

When $n$ has non-zero entries, then the rate at which a measurement is innovative for a class $i$ is described in Section 2.3.4. As an example, a transition occurs from $n = (3, 0, 5, 7)$ at rate $2\mu/N$ to $(0, 3, 5, 7)$ due to a measurement being innovative for class 0 clients.

The total rate at which a measurement is innovative for class $i$ clients is $\bar{\mu}_i = \mu(1 - i/N)$, where $1 - i/N$ is the probability that the new measurement does not belong to $C_i$.

Using Theorem 2.5, the steady-state distribution of the system under the $D$ schedule is as follows.
Corollary 2.1. The steady-state distribution of the system under the \( D \) schedule is:

\[
\Upsilon^D(n) = \prod_{i=0}^{s-1} \left( \frac{\lambda_a}{\lambda_a + \mu(1 - i/N)} \right)^{n_i} \prod_{i=0}^{s-1} \left( \frac{\mu(1 - i/N)}{\lambda_a + \mu(1 - i/N)} \right)^{n_i},
\]

where \( n_i \geq 0 \) for \( 0 \leq i \leq s - 1 \).

2.3.8 Performance Measures - Decentralized Broadcasting

Using Corollary 2.1, we derive the following performance measures of the system under broadcasting schedule \( D \).

Corollary 2.2. Under the \( D \) schedule, the expected number of clients waiting in the system is:

\[
\mathbb{E}[L^D_s] = \sum_{i=0}^{s-1} \frac{\lambda_a}{\mu(1 - i/N)}.
\]

Proof. Direct result from Corollary 2.1.

Corollary 2.3. Under the \( D \) schedule, the expected retrieval time of \( s \) measurements from distinct sensors for a client is:

\[
\mathbb{E}[W^D_s] = \sum_{i=0}^{s-1} \frac{1}{\mu(1 - i/N)}.
\]  \( \quad (2.17) \)

Proof. Direct result from Little’s Law and Corollary 2.2.

Corollary 2.4. Under the \( D \) schedule, the expected length of the busy period is:

\[
\mathbb{E}[K^D_s] = \lambda^{-1}_a \left( \prod_{i=0}^{s-1} \frac{\mu(1 - i/N)}{\lambda_a + \mu(1 - i/N)} \right)^{-1} - 1.
\]

Proof. Direct result from Corollary 2.1.

2.3.9 Optimal Centralized Broadcasting

Consider the following centralized broadcasting schedule, \( O \): at an exponential rate \( \mu \), a measurement is broadcasted only from a sensor whose measurement is innovative for the client waiting the longest in the system. If multiple such sensors exist, then only one of them is scheduled at random for broadcasting. Figure 2.7 shows an example of a realization of the system under the \( O \) schedule.
2.3 Part II: Data Retrieval Time under Wireless Sensor Broadcasting

2.3.10 Analysis of Centralized Broadcasting Schedule $O$

We first make the following claims about the centralized broadcasting schedule described above.

Claim 2.5. Under the $O$ schedule, every measurement broadcast is innovative for all clients present in the system.

Proof. From the fact that each measurement broadcast is innovative for the client waiting the longest in the system (by initial assumption on $O$ schedule) and from Claim 2.2.

Claim 2.6. The $O$ schedule is an optimal broadcasting schedule with respect to the retrieval time of a reliable estimate $\overline{X}$ for a client and expected number of clients in the system.

Proof. From Claim 2.5.

Claim 2.7.

i) All broadcasting schedules that select for every $s$ consecutive broadcasts distinct sensors are optimal with respect to the retrieval time of a reliable estimate $\overline{X}$.

ii) The round-robin broadcasting schedule, which schedules sequentially the $N$ sensors for broadcasting, is also an optimal schedule since any $s$ consecutive broadcasts are from distinct sensors.

From Claim 2.5, all clients in the system receive, upon a broadcast, an innovative measurement. Thus, at rate $\mu$, class $i$ clients change class to $i + 1$, $0 \leq i \leq s - 1$. Clearly, class $s - 1$ clients leave the system.

The $O$ schedule is characterized by the Markov chain introduced in Section 2.3.4, under the following assumptions. The rate $\mu_i = \mu$ for $i = \max\{j | n_j >$
0, 0 ≤ j ≤ s − 1}. For j > i, µ_j = 0 since n_j = 0. Also, µ_j = 0 for j < i (since a broadcast is always innovative for all clients). As an example, consider n = (0, 1, 3, 5). Then, at rate µ a transition occurs from n to (0, 0, 1, 3). Also at rate µ a transition occurs from (1, 3, 5, x) to n, where x > 0.

Using Theorem 2.5, with the observation that \( \bar{\mu}_i = \mu \), which is the rate at which class i clients receive an innovative measurement, the steady-state distribution of the system under the \( O \) schedule is as follows.

**Corollary 2.5.** The steady-state distribution of the system under the schedule \( O \) is:

\[
\Upsilon^O(n) = \left( \frac{\lambda_a}{\lambda_a + \mu} \right)^s \sum_{i=0}^{s-1} n_i \frac{\mu^s}{(\lambda_a + \mu)^s},
\]

where \( n_i ≥ 0, 0 ≤ i ≤ s − 1 \).

### 2.3.11 Performance Measures - Centralized Broadcasting

From Corollary 2.5, we next provide the following performance measures for the system under centralized broadcasting.

**Corollary 2.6.** Under the \( O \) schedule, the expected number of clients waiting in the system is:

\[
\mathbb{E}[L^O_s] = \frac{\lambda_a s}{\mu}.
\]

**Proof.** From Corollary 2.6.

**Corollary 2.7.** Under the \( O \) schedule, the expected time of retrieving \( s \) measurements from distinct sensors for a client is:

\[
\mathbb{E}[W^O_s] = \frac{s}{\mu}.
\]

**Proof.** The result is a direct consequence of Little’s Law and Corollary 2.6. An alternative proof is as follows. From Claim 2.5, each broadcasted measurement is collected by all clients in the system. Moreover, each measurement is broadcasted at an exponential rate \( \mu \). Thus, the expected time each client waits to receive an innovative measurement is \( 1/\mu \). Since each client needs \( s \) distinct measurements, the expected retrieval time for any client to collect \( s \) measurements is Erlang-s distributed with mean \( s/\mu \). This concludes the proof.

\( \square \)
2.4 Conclusions

**Claim 2.8.** Under the centralized broadcasting schedule $O$, there are, in expectation, $\frac{\lambda a}{\mu}$ class $i$ clients present in the system and their expected residual retrieval time is $\frac{s-1}{\mu}$.

*Proof.* From Corollary 2.5 and Corollary 2.7.

**Corollary 2.8.** Under the $O$ schedule, the expected length of the busy period of the system is:

$$E[K_s^O] = \lambda_a^{-1} \left( \left( \frac{\lambda_a + \mu}{\mu} \right)^s - 1 \right).$$

*Proof.* From Theorem 2.8 and Corollary 2.5.

2.3.12 Decentralized vs. Centralized Broadcasting

The next lemma shows that for large-size sensor networks, the $D$ broadcasting schedule converges to the $O$ broadcasting schedule.

**Lemma 2.2.** For $1 \leq s \leq N$,

$$\lim_{N \to \infty} E[W_s^D] = E[W_s^O].$$

*Proof.* From Corollaries 2.3 and 2.7.

2.4 Conclusions

In this chapter we investigated the time for a client to retrieve a reliable estimate of an attribute monitored by a wireless sensor network.

In Part I of this chapter, we considered a time-slotted system where sensors follow an asleep/awake cycle. When awake, the sensors transmit their measurements according to decentralized and centralized and are contending for the medium. We analyzed the retrieval time of a reliable estimate for a client arriving at the system. We showed that the optimal centralized schedule has a lower expected retrieval time than the decentralized schedule. However, the centralized schedule assumes information on the asleep/awake status of the which is difficult to achieve in reality. Nonetheless, we showed that the decentralized schedule, which requires no coordination between the sensors is a constant larger than the optimal centralized schedule when the sampling set of sensors $N$ is sufficiently large.
Additionally, we showed that for large $N$, the retrieval time of a reliable estimate does not depend on the probability of transmitting or being awake. This demonstrates that transmission schedules for wireless sensor networks should be designed jointly with the intended data collection task.

In Part II of this chapter we considered a continuous-time system where multiple clients are interested in retrieving a reliable estimate from the wireless sensor network. Moreover, clients broadcast their measurements according to a general schedule. We introduced a new type of multi-class queueing system for broadcasting, where the clients of the same class receive simultaneously service. We determined closed-form expressions for the steady-state distribution of the system. As special cases, we considered centralized and decentralized sensor broadcasting schedules. We determined closed-form expressions for the steady-state distribution of the system under the centralized and decentralized broadcasting considered. As a consequence, we determined the expected time a client retrieves a reliable estimate from the sensors, the expected number of clients in the system and the expected length of the busy period.
2.5 Appendix

In the proof of Theorem (2.5) we used that (2.14) follows from (2.15) and (2.16). Below we give a detailed proof. We start by adding (2.15) and (2.16).

\[
\prod_{i=0}^{s-3} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_i} \right)^{n_i+1} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_{s-1}} \right)^{n_{s-1}} \cdot \left( \lambda_a + \bar{\mu}_{s-2} - \frac{(\lambda_a + \bar{\mu}_{s-1})^{n_{s-1}+1}}{(\lambda_a + \bar{\mu}_{s-2})^{n_{s-1}}} \right) \gamma_s \\
+ \prod_{i=0}^{s-2} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_i} \right)^{n_i+1} (\lambda_a + \bar{\mu}_{s-1}) \gamma_s \\
= \prod_{i=0}^{s-3} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_i} \right)^{n_i+1} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_{s-1}} \right)^{n_{s-1}} \cdot (\lambda_a + \bar{\mu}_{s-2}) \gamma_s \\
- \prod_{i=0}^{s-3} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_i} \right)^{n_i+1} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_{s-2}} \right)^{n_{s-1}} (\lambda_a + \bar{\mu}_{s-1}) \gamma_s \\
+ \prod_{i=0}^{s-2} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_i} \right)^{n_i+1} (\lambda_a + \bar{\mu}_{s-1}) \gamma_s \\
= \prod_{i=0}^{s-3} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_i} \right)^{n_i+1} \left( \frac{\lambda_a}{\lambda_a + \bar{\mu}_{s-1}} \right)^{n_{s-1}} \cdot (\lambda_a + \bar{\mu}_{s-2}) \gamma_s,
\]

where the last equality follows from the observation that the second and third terms in (2.18) are equal. The result follows.
3.1 Introduction

Wireless sensor networks are often deployed at random in an area of interest. This can be the case, for instance, of sensors that detect and track fires in forests [109,112]. These sensors are, in general, randomly spread over the terrain and take measurements over attributes such as temperature, humidity, CO₂, etc. Moreover, reliable data from these sensors often needs to be retrieved within some predefined time window to timely localize fires and, if required, to allow for intervention.

In this chapter, we extend the analysis of the problem of measurement retrieval from wireless sensors, which was introduced in Chapter 2, by considering the energy consumption of the sensors that transmit measurements. More specifically, we consider the situation where wireless sensors are placed in the plane according to a random spatial process. We assume that these sensors have noisy measurements of one and the same attribute (for instance, temperature, CO₂ level). A sink, at a random location in the plane, is interested in retrieving measurements from the sensor network. The sink could be viewed, for instance, as a user with a mobile device that enters the area in which the sensor network is deployed and retrieves sensor measurements. Similar to the setting of Chapter 2, we are interested that the sink retrieves a reliable estimate of an attribute by combining the measurements of a sufficiently large set of sensors. The reliability of an estimate is determined using the model proposed in Chapter 2, Section 2.1.1. In addition, we are interested in retrieving a reliable estimate within a predefined time window and with minimum energy consumption for the sensors which transmit their measurements to the sink.

The network is assumed to consist of sensors that harvest energy from the environment. More precisely, we assume that the sensors have batter-
ies with limited storage capacities and that the energy gets replenished over time according to a random process. This implies that the sensors will not always have energy available to transmit their measurements to the sink. When a sensor does not have enough energy to transmit or receive messages, it enters a sleep state. Once sufficient energy is harvested, the sensor enters the awake state and is able to transmit messages again. Since the sensors perform additional tasks (for example, transmitting measurements to other sinks, obtaining new measurements), a new sleep cycle can be entered even if no messages have been transmitted to the sink. Summarizing, the sensors follow an asleep/awake pattern according to a random process and measurements can be obtained by the sink only if a sensor is awake.

We further assume that a beaconing mechanism is in place from which the sink can infer which sensors are awake in each time slot. The sink can then request a measurement from an awake sensor. Moreover, to avoid retrieving multiple times the same measurement from the same sensor, which could lead to a biased estimate of the attribute, the sink will request a measurement only from an awake sensor which is distinct from the ones that have already transmitted their measurements.

Since the sensors have limited energy reserves, in order to prolong the lifetime of the sensor network, the sink aims to minimize the energy which the sensors use for transmission. We will consider the general setting in which the energy of a sensor is an increasing function of the distance between the sensor and the sink. Therefore, the sink aims to retrieve measurements from sensors which are as close as possible to it. In addition, the sink has a predefined time window within which it can retrieve sufficient measurements for a reliable estimate. This time constraint ensures that the estimate is based on the most up-to-date measurements of the attribute available in the sensor network.

The sink could reduce the energy consumption of the sensors by waiting for close sensors to become awake and request them to transmit their measurements. However, the time until some of these sensors become awake could exceed the predefined time window within which measurements can be retrieved. Therefore, the sink could retrieve too few measurements, based on which a reliable estimate cannot be retrieved. Alternatively, to retrieve sufficient sensor measurements within a predefined time window, the sink could request the sensors to transmit as soon as they become awake, without taking into account how far these sensors are from the sink. In this case, however, the energy consumption needed to retrieve a reliable estimate could be high.
3.1 Introduction

Our interest is to determine which sensors should be scheduled for transmission and when, so that the sink retrieves sufficient measurements to compute a reliable estimate within a predefined time window and with minimum energy consumption for the sensors.

The problem of reliable and energy-efficient data retrieval for wireless sensor networks has been addressed in [8, 9, 15, 70, 71, 89, 108, 109, 112]. In most studies, however, the problem of energy-efficient data retrieval does not consider both aspects related to time deadlines for data retrieval, as well as the deployment of the sensors in the plane, as it is the case of the model considered in this chapter. In [112] an architecture for a wireless sensor network used for forest fire detection is described. Similar to the setting proposed in this chapter, the sensors are periodically requested to send measurements to an aggregation point. Based on the sensed data received, fire alarms can be triggered. In [109], cluster heads retrieving sensor measurements are dynamically selected such that the energy used for data transmission and retrieval is minimized. However, no constraints on the time during which the data retrieval must be done, are considered in [109, 112], as in the model employed in this chapter. In [9], a survey on energy conservation schemes in wireless sensor networks is provided. In [15], energy minimization is considered in a wireless sensor network with multiple sinks and multiple aggregation nodes. Using stochastic geometry, the expected energy consumption associated with an optimal organizational hierarchy is determined. In contrast to the model considered in this chapter, where we retrieve measurements from a subset of sensors, in [15] the sink compresses the data from all sensors. Moreover, the authors do not consider any time constraints for the data compression at the sink. In [71], energy harvesting nodes are considered. These nodes transmit data to an access point in the up-link only if they have sufficient energy stored in a buffer. The authors determine the limiting distribution of the stored energy in finite and infinite energy buffers. In [108], a collision-free schedule for data aggregation in wireless sensor networks is proposed based on maximal independent sets. A bound on the delay of the schedule is provided. In [8], the problem of reliable and energy efficient data retrieval from a sparse sensor network to a mobile data collector, is considered. The authors consider a limited contact time within which data can be transmitted from the sensors to the collector. This contact time depends on the time the mobile collector is within the transmission range of a sensor. In [89, 110, 111], the authors provide an energy-efficient transmission schedule of a fixed number $B$ of packets from one transmitter to one receiver. Similar to our setting, the packets must be
transmitted within a predefined time period of $T$ units. The schedule provided in [89] specifies the time slots when the transmitter should send a packet to the receiver or be silent, whereas in [110,111] an optimal transmission rate is derived. In addition, the energy consumption of the transmitter is assumed to be a function of the state of the transmission channel. The schedules take into account the state of the transmission channel, which influences the energy used to successfully transmit a packet. In our setting, we derive energy-efficient transmission schedules while taking into account the asleep/awake state of the sensors. Also, the energy consumption is a function of the distance over which transmissions occur.

In this chapter we consider the problem of retrieving a reliable estimate of an attribute from a wireless sensor network within a fixed time window and with minimum energy consumption for the sensors. Firstly, we derive a closed-form expression for the expected energy consumption of sensors when the sink requires measurements according to a Greedy schedule. This expression is general in the sense that it does not depend on the random process according to which the sensors are placed in the plane or on the function of the energy consumption. We also determine an upper bound for the maximum expected distance over which a sensor transmits under this Greedy schedule. We apply these results for sensors located in the plane according to a homogeneous spatial Poisson process. Secondly, we formulate a stochastic dynamic programming model (SDP) to determine a transmission schedule with general time constraints such that a reliable estimate is retrieved with minimum sensor energy consumption. Lastly, we develop one heuristic and an offline, optimal schedule for the retrieval of sensor measurements and compare numerically the energy consumption of the two schedules with the SDP schedule. The offline, optimal schedule and the heuristic provide a lower and upper bound, respectively, for the energy consumption achieved under the SDP schedule.

The remainder of this chapter is organized as follows. In Section 3.2 we formulate the problem statement. In Section 3.3 we compute the minimum expected energy consumption under a Greedy sensor transmission schedule. In Section 3.4 we apply our findings from Section 3.3 for sensors placed in the plane according to a homogeneous Poisson spatial process. In Section 3.5 we determine a sensor transmission schedule with general time constraints. We also consider a heuristic and an off-line, optimal sensor transmission schedule that minimize energy consumption while ensuring that a reliable estimate is retrieved within a predefined time window. In Section 4.4 we provide numerical results for the schedules considered in Section 3.5.
In Section 4.5 we discuss the results and provide conclusions.

### 3.2 Model and Problem Statement

We consider wireless sensors located in the plane according to a random spatial process. Each sensor has a measurement of an attribute $\theta$. The measurements are subject to additive normally distributed noise with variance $\sigma^2$, i.e.,

$$ X_i \sim N(\theta, \sigma^2). $$

The noise is independent and identically distributed across the sensors.

We assume that time is slotted, where a time slot is defined as the time it takes for a sensor to transmit a measurement to the sink. The sink, located at a random location in the plane, is interested in retrieving a reliable estimate $\bar{X}$ of $\theta$ from the sensors. The model for reliability of an estimate of an attribute has been introduced in Chapter 2. For completeness, we restate it here. We assume an estimate is reliable if the variance of $\bar{X}$ is below a threshold $H$. Moreover, the estimate must be retrieved within a fixed time window of $t$ time slots. The estimate can be obtained by retrieving, within $t$ time slots, an arbitrary set of $s$ measurements from $s$ distinct sensors such that:

$$ \text{Var}(\bar{X}) = \text{Var}(\frac{1}{s} \sum_{i=1}^{s} D_i) = \frac{1}{s^2} \sum_{i=1}^{s} \text{Var}(X_i) = \frac{\sigma^2}{s} < H. $$

We consider $s = \lceil \frac{\sigma^2}{H} \rceil$. Since at most one measurement can be obtained in each time slot, the time constraint must satisfy $t \geq s$.

Every time slot, the sensors are awake independently at random with probability $p$, $0 < p < 1$, and asleep with probability $q = 1 - p$. The probability of being awake is independent over time and across the sensors. We say that a sensor is eligible to transmit if the sensor is awake and has not transmitted its measurements to the sink in the previous time slots. Eligibility prevents the sink from receiving multiple times the same measurement from the same sensor.

Let the sensors be sorted increasingly according to their distance to the sink. We assume that the sink knows the position of the sensors in the plane. Moreover, every time slot, the sink observes which sensors are eligible as a beaconing mechanism is in place. Therefore, a sensor is requested by the sink to transmit only if it is eligible.

Let $i$ denote the sensor that is the $i$-th closest sensor to the sink. Let $\delta_i$ denote the distance between the sink and sensor $i$. We assume that a
sensor at any random location in the plane can transmit its measurement to the sink at a corresponding energy. Let $e(\delta_i)$ denote the energy used by sensor $i$ to transmit to the sink. Let $\epsilon(i) = \mathbb{E}[e(\delta_i)]$, where the expectation is taken over the randomness of the random spatial process according to which the sensors are placed in the plane. Let $\xi^t_x$ denote the probability that a set of sensors $x = \{x_1, \ldots, x_s\}, x \subset \mathbb{N}^+$ transmits to the sink within $t$ time slots, under some sensor transmission schedule. Then, the expected energy consumption used to retrieve a reliable estimate of an attribute within $t$ time slots is:

$$
\mathbb{E}[P^t_s] = \sum_{x \subset \mathbb{N}^+} \left( \sum_{x_i \in x} \epsilon(x_i) \right) \xi^t_x,
$$

where $1 \leq i \leq s$ and the expectation is taken over the randomness in the random spatial process according to which the sensors are located in the plane and the randomness in the asleep/awake state of the sensors.

We are interested in developing sensor transmission schedules that minimize the expected energy consumption $\mathbb{E}[P^t_s]$ of the transmitting sensors while ensuring that a reliable estimate is retrieved within a fixed time window of $t \geq s$ time slots.

### 3.3 Energy Consumption and Tight Time Constraints: Greedy Schedule

In this section we determine a closed-form expression for the total expected energy used by the sensors to transmit $s$ measurements to the sink within $t = s$ time slots when an online Greedy transmission schedule is considered. We also determine an upper bound for the maximum distance over which a sensor transmits under this Greedy schedule.

#### 3.3.1 Greedy Schedule

We consider the following, intuitively appealing, online Greedy schedule. The sink, which has knowledge about the location of the sensors in the plane, observes every time slot which sensors are eligible and sends a request in each slot to the closest and eligible sensor to transmit its measurement. See Figure 3.1 for an example of a realization of Greedy.

Recall that the sink aims to minimize the energy consumption of the sensors. Given that every time slot the sink needs to request a sensor to transmit so that, in the end, it retrieves exactly $s$ measurements, and given the online
3.3 Energy Consumption and Tight Time Constraints: Greedy Schedule

Figure 3.1: Example of a realization of Greedy schedule, where $s = 4$ measurements from eligible sensors need to be retrieved by the sink within $t = 4$ time slots. The sensors are sorted according to their distance to the sink. Every time slot, the closest eligible sensor is requested to transmit to the sink. A black, filled circle indicates an awake sensor, an empty circle indicates an asleep sensor and a rectangle indicates a non-eligible sensor.
setting, the Greedy schedule results in the lowest possible energy consumption needed to retrieve \( s \) measurements in \( t = s \) time slots.

### 3.3.2 Preliminaries

Let \( x = \{x_1, x_2, \ldots, x_u\} \) be a set of size \( u \), \( x \subset \mathbb{N}^+ \). Let the vector \( \tau(x) \) be a permutation of the set \( x \). Let \( \tau(x)_i \) denote the element of \( \tau(x) \) at position \( i \). Let \( S_x \) denote the set of all permutations of \( x \). Let \( D_u = \{ \tau(x) \in S_x \mid x \subset \mathbb{N}^+, |x| = u \} \). Let \( O_u = \{ \tau(x) \in D_u \mid \tau(x)_1 < \tau(x)_2 < \ldots < \tau(x)_u \} \). For \( k \geq u \), let \( D_u(k) = \{ \tau(x) \in S_x \mid x \subset \{1, 2, \ldots, k\}, |x| = u \} \) and \( O_u(k) = \{ \tau(x) \in D_u(k) \mid \tau(x)_1 < \tau(x)_2 < \ldots < \tau(x)_u \} \).

The following definition is well-known in literature (see, for instance, [18]).

**Definition 3.1.** We say that \( \tau(x)_i \) is in inversion with \( \tau(x)_j \) in the permutation \( \tau(x) \) if \( i < j \) but \( \tau(x)_i > \tau(x)_j \). We call the pair \( (\tau(x)_i, \tau(x)_j) \) an inversion.

We also find it convenient to consider the following two definitions.

**Definition 3.2.** The inversion number of \( \tau(x)_j \), denoted by \( v(\tau(x)_j) \), is \( v(\tau(x)_j) = \sum_{i=1}^{j-1} 1_{\tau(x)_i > \tau(x)_j} \).

**Definition 3.3.** The disorder of a permutation \( \tau(x) \) of a set \( x \) of size \( u \), denoted by \( \eta(\tau(x)) \), is \( \eta(\tau(x)) = \sum_{i=1}^{u} v(\tau(x)_i) \).

By definition, \( v(\tau(x)_j) \) is the number of elements of \( \tau(x) \) up to position \( j \) which are in inversion with \( \tau(x)_j \). Also, the disorder of a permutation is the number of inversions it contains. Clearly, for a set \( x \) with \( |x| = u \), \( 0 \leq \eta(\tau(x)) \leq u(u-1)/2 \).

As an example, consider the permutation \( \tau(x) = (3, 1, 5, 2, 4) \) and \( \tau(x)_5 = 2 \). Then, \( v(\tau(x)_5) = 2 \), because 3 and 5 are in inversion with 2 and \( \eta((3, 1, 5, 2, 4)) = 4 \) since \((3, 1), (3, 2), (5, 2)\) and \((5, 4)\) are all the inversions of this permutation.

Let \( N_i(x) \) be the number of permutations with disorder \( i \) of a set \( x \) of size \( u \), i.e.,

\[
N_i(x) = \sum_{\tau(x) \in S_x} 1_{\eta(\tau(x)) = i}.
\]

From [18, Thm 2.3] we have the following result.

**Theorem 3.1.** [18, Thm 2.3] For \( u \geq 1 \),

\[
\sum_{i=0}^{u(u-1)/2} N_i(x) \cdot y^i = \prod_{j=0}^{u-1} (1 + y + \ldots + y^j).
\]
According to Theorem 3.1, \( N_i(x) \) is the coefficient of \( y^i \), \( 0 \leq i \leq \frac{n(n-1)}{2} \), in the polynomial \( \prod_{j=0}^{u-1} (1 + y + \ldots + y^j) \).

For ease of notation, let \( C_s = p_s q_s - s(s+1)/2 \prod_{m=0}^{s-1} (1 + q + \ldots + q^m) \).

### 3.3.3 Expected Energy Consumption under Greedy

In this section we determine the total expected energy consumption of the sensors transmitting under Greedy.

Let \( x \) be the set of sensors that transmit under Greedy, \( x \subset \mathbb{N}^+ \) and \( |x| = s \). Let \( \tau(x) \) be the sequence in which the set \( x \) transmits under Greedy, i.e., sensor \( \tau(x)_i \) transmits at slot \( i \). Let \( p_x \) be the probability that the set \( x \) of sensors transmits under Greedy. Let \( p_{\tau(x)} \) be the probability that the set \( x \) of sensors transmits in the sequence \( \tau(x) \) under Greedy. Clearly, \( p_x = \sum_{\tau(x) \in S_x} p_{\tau(x)} \).

First, we state the following lemma, used in the proof of Theorem 3.2.

**Lemma 3.1.** The probability that a set \( x \), \( |x| = s \), transmits under Greedy in a sequence \( \tau(x) \) is:

\[
\bar{p}_{\tau(x)} = p^s q^{s(s+1)/2 + \eta(\tau(x))} \prod_{i=1}^{s} q^{x_i}.
\]

**Proof.** The probability that the set \( x \) of sensors transmits under Greedy in a sequence \( \tau(x) \), with \( \eta(\tau(x)) = i \), is the probability that, at slot \( j \), \( 1 \leq j \leq s \), sensor \( \tau(x)_j \) is awake and all \( \tau(x)_{i-1} \) sensors, which are closer to the sink than \( \tau(x)_j \), are asleep, except all sensors \( \tau(x)_i \) with \( 1 \leq i < j \) and \( \tau(x)_i < \tau(x)_j \).

Let \( N_c \) be the number of sensors \( \tau(x)_i \) with \( 1 \leq i < j \) and \( \tau(x)_i < \tau(x)_j \). Clearly, \( 0 \leq N_c < \tau(x)_j \).

Let \( N_f \) be the number of sensors \( \tau(x)_i \) with \( 1 \leq i < j \) and \( \tau(x)_i > \tau(x)_j \). It is easy to see that \( j - 1 = N_c + N_f \).

By Definition 2, \( N_f = v(\tau(x)_j) \), with \( 0 \leq v(\tau(x)_j) \leq j - 1 \). Then, \( N_c = j - 1 - v(\tau(x)_j) \). Therefore,

\[
\mathbb{P}(\tau(x)_j \text{ at slot } j \mid \tau(x)_i \text{ at slot } i, \forall i < j) = pq^{(\tau(x)_j-1)-N_c} = pq^{(\tau(x)_j-j+v(\tau(x)_j))}.
\] (3.2)
From Equation (3.2), the probability that \( x \) transmits in a sequence \( \tau(x) \) under Greedy is:

\[
\bar{p}_{\tau(x)} = \prod_{j=1}^{s} pq^{\tau(x)_j - j + \nu(\tau(x)_j)} = p^s q^{x_1 + \ldots + x_s - s(s+1)/2 + \sum_{j=1}^{s} \nu(\tau(x)_j)}.
\]

The result follows from Definition 3. 

We next determine the probability of a set \( x \) of sensors, where \( |x| = s \), transmitting under Greedy, which takes into account all possible sequences in which this set of sensors could transmit.

**Theorem 3.2.** The probability that a set \( x \) of size \( s \) of sensors transmits under Greedy is:

\[
p_x = C_s \prod_{i=1}^{s} q^{x_i}.
\]

**Proof.** The probability that a set \( x \) of size \( s \) transmits under Greedy is expressed as:

\[
p_x = \sum_{\tau(x) \in S_x} \bar{p}_{\tau(x)}
\]

\[
= \sum_{i=0}^{s(s-1)/2} \left( \sum_{\tau(x) \in S_x} \bar{p}_{\tau(x)} \mathbf{1}_{\eta(\tau(x)) = i} \right)
\]

\[
= p^s q^{-\frac{s(s+1)}{2}} \prod_{i=1}^{s} q^{x_i} \sum_{i=0}^{s(s-1)/2} N_i(x) q^i
\]

\[
= p^s q^{-\frac{s(s+1)}{2}} \prod_{i=1}^{s} q^{x_i} \prod_{m=0}^{s-1} \sum_{j=0}^{m} q^j,
\]

where Equation (3.5) follows from Lemma 3.1, Equation (3.6) follows from Theorem 3.1 and from the definition of \( C_s \). 

In the next theorem we determine the expected energy consumption under Greedy. First we introduce two technical lemmas.

**Lemma 3.2.** For \( s \geq 1, a \geq 1 \) and \( x \subset \mathbb{N}^+, |x| = s \),

\[
\sum_{\tau(x) \in D_s} \epsilon(\tau(x)_1) p_x = (s - 1)! \sum_{\tau(x) \in O_s} \left( \sum_{i=1}^{s} \epsilon(\tau(x)_i) \right) p_x.
\]
3.3 Energy Consumption and Tight Time Constraints: Greedy Schedule

Proof. Appendix, Section 3.8.1.

Lemma 3.3. For \( s > 1, a \geq 1 \) and \( x \subset \mathbb{N}^+, |x| = s \),

\[
\sum_{\tau(x) \in \mathcal{D}} \epsilon(\tau(x)_1) \prod_{i=1}^{s} q^{\tau(x)_i} = (s-1)! \sum_{\tau(x)_1=1}^{\infty} \epsilon(\tau(x)_1) q^{\tau(x)_1} \sum_{j=1}^{s} \frac{(-1)^{j+1} q \frac{(s-j)(s-j+1)}{2} q^{(j-1)\tau(x)_1}}{\prod_{i=1}^{s-j} (1 - q^i)}.
\]

Proof. Appendix, Section 3.8.2.

Theorem 3.3. The total expected energy consumption of the sensors under Greedy is:

\[
\mathbb{E}[P_s^s] = C_s \sum_{j=1}^{s} \frac{(-1)^{j+1} q \frac{(s-j)(s-j+1)}{2} q^{(j-1)\tau(x)_1}}{\prod_{i=1}^{s-j} (1 - q^i)} \sum_{k=1}^{\infty} \epsilon(k) q^{ik}.
\]

Proof. Let \( x \subset \mathbb{N}^+, |x| = s \), be a set of sensors transmitting under Greedy. The expected energy used for transmission under Greedy can be written as:

\[
\mathbb{E}[P_s^s] = \sum_{\tau(x) \in \mathcal{D}_s} \left( \sum_{i=1}^{s} \epsilon(\tau(x)_i) \right) p_x \tag{3.7}
\]

\[
= \frac{1}{(s-1)!} \sum_{\tau(x) \in \mathcal{D}_s} \epsilon(\tau(x)_1) p_x \tag{3.8}
\]

\[
= \frac{C_s}{(s-1)!} \sum_{\tau(x) \in \mathcal{D}_s} \epsilon(\tau(x)_1) \prod_{i=1}^{s} q^{\tau(x)_i} \tag{3.9}
\]

\[
= \frac{C_s}{(s-1)!} (s-1)! \sum_{\tau(x)_1=1}^{\infty} \epsilon(\tau(x)_1) q^{\tau(x)_1} \sum_{j=1}^{s} \frac{(-1)^{j+1} q \frac{(s-j)(s-j+1)}{2} q^{(j-1)\tau(x)_1}}{\prod_{i=1}^{s-j} (1 - q^i)}, \tag{3.10}
\]

where (3.8) follows from Lemma 3.2; (3.9) follows from substituting \( p_x \), according to Theorem 3.2 with the observation that \( \sum_{i=1}^{s} x_i = \sum_{i=1}^{s} \tau(x)_i \); (3.10) follows from Lemma 3.3. The result now follows from substituting \( \tau(x)_1 \) by \( k \). 

\( \Box \)
Theorem 3.3 contains an infinite sum, which depends on a general function $\epsilon(k) > 0$ denoting the expected energy consumption of sensor $k$ to transmit to the sink. This function can depend on the distribution of the sensors in the plane, on the transmission environment, etc. In Section 3.4, the sensors are assumed to be located in the plane according to a homogeneous Poisson process and a specific function for the energy consumption is considered. This leads to a closed-form expression for the energy consumption of the transmitting sensors.

3.3.4 Maximum Expected Transmission Range under Greedy

In this section we determine the probability that the furthest sensor from the sink, which transmits under Greedy, is $k$, with $k \geq s \geq 1$. Since the energy consumption of a sensor is assumed to be an increasing function of the distance between that sensor and the sink [3], the maximum transmission range gives an indication on the maximum energy consumption of a single sensor when the transmission is done according to Greedy.

We begin by stating the following technical lemma, which will be used in the proof of Theorem 3.4.

Lemma 3.4. For $k \geq s \geq 1$,

$$\sum_{\tau(z) \in D_{k-s}(k-1)} \prod_{i=1}^{k-s} q^{-\tau(z)_i} = (k-s)!q^{-(k-s)s-\frac{(k-s-1)(k-s)}{2}} \prod_{i=1}^{k-s} \frac{1 - q^{s+i-1}}{1 - q^i}.$$ 

Proof. Appendix, Section 3.8.3

We next determine the probability that sensor $k$, with $k \geq s$, is the furthest sensor from the sink that transmits under Greedy.

Theorem 3.4. Given a set of sensors $x \subset \mathbb{N}^+$, $|x| = s$ transmitting under Greedy, the probability that sensor $k \in x$, with $k \geq s \geq 1$, is the furthest sensor from the sink that transmits, is:

$$\mathbb{P}(\max(x) = k) = q^{k-s}(1 - q^s) \prod_{i=k-s+1}^{k-1} (1 - q^i).$$

Proof. By Theorem 3.2 where, without loss of generality, we assume that
Lemma 3.4; (3.15) follows from substituting \( x \) by \( \tau \) where (3.13) follows from writing 
\[
\sum_{1 \leq x_i < k, 1 \leq i < s} x_i = \max(x)
\]
to take into account only once the values of \( x \). Thus, \( z = \{1, 2, \ldots, k - 1\}\{x_1, x_2, \ldots, x_{s-1}\} \). Since we consider \( \tau(z) \in \mathbb{D}_{k-s}(k-1) \), i.e. all permutations of the set \( z \), we divide by \((k-s)!\) to take into account only once the values of \( z \); (3.14) follows from Lemma 3.4; (3.15) follows from substituting \( C_s \).

To obtain the result, we next consider two cases. First, consider the case 
\[
k < 2s.
\]
Then,
\[
\begin{align*}
\mathbb{P}(\max(x) = k) &= p^s q^{k-s} \prod_{j=0}^{s-1} \left( \sum_{m=0}^{j} q^m \right) \prod_{j=k-s}^{s-1} \left( \sum_{m=0}^{j} q^m \right) \prod_{i=1}^{k-s} \frac{1 - q^{s+i-1}}{1 - q^i} \\
&= (1 - q)^{2s-k} q^{k-s} \prod_{j=k-s}^{s-1} \left( \sum_{m=0}^{j} q^m \right) \prod_{i=1}^{k-s} (1 - q^{s+i-1}) \\
&= q^{k-s} \prod_{j=k-s}^{s-1} \left( (1 - q) \cdot \left( \sum_{m=0}^{j} q^m \right) \right) \prod_{i=1}^{k-s} (1 - q^{s+i-1}) \\
\end{align*}
\]
\begin{equation}
\begin{aligned}
q^{k-s} \prod_{j=k-s}^{s-1} (1 - q^{j+1}) \prod_{i=1}^{k-s} (1 - q^{s+i-1}) \\
= q^{k-s} (1 - q^s) \prod_{j=k-s}^{s-1} (1 - q^{j+1}) \prod_{j=s}^{k-2} (1 - q^{j+1}) \\
= q^{k-s} (1 - q^s) \prod_{i=k-s+1}^{k-1} (1 - q^i),
\end{aligned}
\tag{3.18}
\end{equation}

where we used the fact that \(1 - q^j = (1 - q) \sum_{m=0}^{j-1} q^m\) in (3.16) and (3.17).

Next, we consider the case \(k \geq 2s\). Then,

\begin{equation}
\begin{aligned}
P(\max(x) = k) &= p^s q^{k-s} \prod_{j=0}^{s-1} \left( \sum_{m=0}^{j} q^m \right) \prod_{i=1}^{k-s} (1 - q^{s+i-1}) \prod_{i=s+1}^{k-s} \frac{1}{1 - q^i} \\
&= p^s q^{k-s} \prod_{j=0}^{s-1} \left( \sum_{m=0}^{j} q^m \right) \prod_{i=1}^{k-s} (1 - q^{s+i-1}) \prod_{i=s+1}^{k-s} \frac{1}{1 - q^i} \\
&= (1 - q)^s q^{k-s} \prod_{i=1}^{k-s} (1 - q^{s+i-1}) \prod_{i=s+1}^{k-s} \frac{1}{1 - q^i} \\
&= q^{k-s} \prod_{i=1}^{k-s} (1 - q^{s+i-1}) \prod_{i=s+1}^{k-s} \frac{1}{1 - q^i} \\
&= q^{k-s} (1 - q^s) \prod_{i=2}^{k-s} (1 - q^{s+i-1}) \prod_{i=s+1}^{k-s} \frac{1}{1 - q^i} \\
&= q^{k-s} (1 - q^s) \prod_{i=s+1}^{k-1} (1 - q^i) \prod_{i=s+1}^{k-s} \frac{1}{1 - q^i} \\
&= q^{k-s} (1 - q^s) \prod_{i=k-s+1}^{k-1} (1 - q^i),
\end{aligned}
\tag{3.19}
\end{equation}

where we used the fact that \(1 - q^j = (1 - q) \sum_{m=0}^{j-1} q^m\) in (3.19). The result follows from (3.18) and (3.20).

Lastly, we make the following simple observation, which will be used in the next section.
**Observation 3.1.** For $k \geq s \geq 1$,

$$\mathbb{P}(\max(x) = k) \leq q^{k-s}(1-q^s)(1-q^{k-1}).$$

In this section we have considered a general spatial random process according to which the sensors are located in the plane. In the following section we apply our results for the specific case when sensors are placed according to a homogeneous spatial Poisson process.

### 3.4 Spatial Poisson Process - Greedy Schedule

In the previous sections we considered a general random spatial process according to which the sensors are placed in the plane and a general function for the energy consumption of a sensor that transmits to the sink.

In this section we determine the total expected energy consumption and the maximum distance over which a sensor transmits under Greedy when the sensors are positioned in the plane according to a homogeneous spatial Poisson process (HPP) with intensity $\lambda$. To do so, we consider the energy used by sensor $u$, $u \in \mathbb{N}^+$, to transmit to the sink to be

$$e(u) = \delta_u^a,$$

where $a \geq 1$ is a fixed parameter. Recall that the sensors have been sorted according to their distance to the sink. Therefore, sensor $u$ is the closest $u$–th sensor to the sink.

The form of the energy function is motivated by, for instance, the minimum power $P = (e^{2G} - 1)\delta^a$ required to transmit at a fixed rate $G$ over a distance $\delta$ given that the capacity of a AWGN (Additive White Gaussian Noise) channel is

$$\frac{1}{2} \log(1 + P\delta^{-a}),$$

where $a$ denotes the path loss exponent of the transmission environment, which in the plane satisfies $a \geq 2$.

The form of the energy function can be also motivated by an example on the time and, therefore, the energy needed to transmit a measurement from a sensor $u$ to the sink when TCP is used. It is known [6,78] that the throughput of TCP is inversely proportional to the round trip delay. The delay is the sum of the processing delay (which we assume to be zero) and the propagation delay, which is proportional to the distance $\delta_u$. Thus, the time needed for
sensor $u$ to transmit its measurement using TCP is $W = c \cdot \delta_u$, where $c$ is a constant and $a = 1$.

Lastly, let $M_s$ be the maximum distance over which a sensor transmits under the Greedy schedule.

We first state the following lemma, which will be used in Corollaries 3.1 and 3.2.

**Lemma 3.5.** For $a \geq 1$, \[
\sum_{u=1}^{\infty} \frac{\Gamma(a/2 + u)}{\Gamma(u)} q^u = \frac{q \Gamma(a/2 + 1)}{(1 - q)^{a/2+1}}.
\]

**Proof.** Using that $\Gamma(a/2 + u) = \int_0^{\infty} y^{a/2+u-1}e^{-y}dy$ (see, for instance, [1]),

\[
\sum_{u=1}^{\infty} \frac{\Gamma(a/2 + u)}{\Gamma(u)} q^u = \sum_{u=1}^{\infty} \int_0^{\infty} \frac{y^{a/2+u-1}e^{-y}q^u}{\Gamma(u)} dy
\]

\[
= \int_0^{\infty} y^{a/2-1}e^{-y} \sum_{u=1}^{\infty} \frac{y^u q^u}{\Gamma(u)} du
\]

\[
= \int_0^{\infty} y^{a/2} q e^{-y(1-q)} dy.
\]

Equation (3.22) follows from interchanging the summation with the integration. Let $f(u, y) = \frac{y^{a/2+u-1}e^{-y}q^u}{\Gamma(u)}$ where $f(u, y) \geq 0, \forall u \geq 0, \forall y \geq 0$. Using that $\int_0^{\infty} f(u, y)dy = \frac{\Gamma(a/2+u)}{\Gamma(u)} q^u < \infty$ and $\sum_{u=1}^{\infty} f(u, y) = y^{a/2}e^{-y(1-q)}q < \infty$, by Fubini’s Theorem (see, for instance, [85], Theorem 8.8), we can interchange the summation with the integration. Equation (3.23) follows from the fact that

\[
\sum_{u=1}^{\infty} \frac{y^u q^u}{\Gamma(u)} = yq \sum_{i=0}^{\infty} \frac{(yq)^i}{i!} = yqe^{yq}.
\]

Using that $\Gamma(m) = k^m \int_0^{\infty} y^{m-1}e^{-ky}dy$, for positive reals $m, k$ (see, for instance, [1]), it follows that

\[
\int_0^{\infty} y^{a/2} e^{-y(1-q)}qdy = q \Gamma(a/2 + 1)(1 - q)^{-a/2-1}.
\]

This concludes the proof. \(\square\)
3.4 Energy Consumption for Greedy: Spatial Poisson Process

**Corollary 3.1.** For sensors placed in the plane according to a homogeneous Poisson process with intensity $\lambda$ and $a \geq 1$, the total expected energy consumption under Greedy is:

$$
\mathbb{E}[P_{s,HPP}^s] = C_s \frac{\Gamma(a/2 + 1)}{(\lambda \pi)^{a/2}} \sum_{j=1}^{s} \frac{(-1)^{j+1} q^{(s-j)(s-j+1)/2}}{s-j \prod_{i=1}^{s-j} (1 - q^i)} \frac{q^j}{(1 - q^j)^{a/2+1}}.
$$

**Proof.** The expected energy consumption used by sensor $k \in x, k \geq 1$ to transmit to the sink under Greedy, where the sensors are located in the plane according to a homogeneous Poisson process, can be written as follows [7]:

$$
\epsilon(k) = \int_{0}^{\infty} u^a d\mathbb{P}(\delta_k \leq u) \quad (3.24)
$$

$$
\epsilon(k) = \int_{0}^{\infty} u^a d \left( 1 - \frac{\Gamma(k, \lambda \pi u^2)}{\Gamma(k)} \right) \quad (3.25)
$$

$$
\epsilon(k) = \frac{\Gamma(a/2 + k)}{(\lambda \pi)^{a/2} \Gamma(k)}, \quad (3.26)
$$

where equation (3.25) follows from the fact that, for a Poisson random variable $X$ with mean $v$,

$$
\mathbb{P}(X < k) = e^{-v} \sum_{i=0}^{k-1} \frac{v^i}{\Gamma(i+1)} = \frac{\Gamma(k, v)}{\Gamma(k)}.
$$

Thus, the distance to the $k$-th closest sensor to the sink, when the sensors are placed in the plane according to a homogeneous Poisson process with intensity $\lambda$, is at most $u$ is

$$
\mathbb{P}(\delta_k \leq u) = 1 - \frac{\Gamma(k, \lambda \pi u^2)}{\Gamma(k)}.
$$

Now, substituting Equation (3.26) in Theorem 3.3,

$$
\mathbb{E}[P_{s,HPP}^s] = C_s \frac{\Gamma(a/2 + 1)}{(\lambda \pi)^{a/2}} \sum_{j=1}^{s} \frac{(-1)^{j+1} q^{(s-j)(s-j+1)/2}}{s-j \prod_{i=1}^{s-j} (1 - q^i)} \frac{q^j}{(1 - q^j)^{a/2+1}}.
$$
where in the last equality we used Lemma 3.5.

Figure 3.2 shows the total expected energy consumption of the sensors transmitting under Greedy when the intensity $\lambda$ of the homogeneous Poisson process, according to which the sensors are placed in the plane, and the path loss exponent of the transmission environment, $a$, vary. As expected, as the intensity with which the sensors are placed in the plane increases, the energy consumption of the transmitting sensors decreases. This is because the distance between the sink and the transmitting sensors is expected to decrease. A similar effect is obtained for the case when sensors are awake with high probability. In this case, the probability of retrieving measurements from sensors, located close to the sink, increases.

### 3.4.2 Expected Maximal Transmission Range for Greedy: Spatial Poisson Process

In the next corollary, let $a = 1$. Note that for $a = 1$, $e(u) = \delta_u$, which is the distance between the sink and the transmitting sensor.

**Corollary 3.2.** For sensors placed in the plane according to a HPP with intensity $\lambda$ and $a = 1$, the expected maximal transmission range under Greedy is bounded as follows:

$$
\mathbb{E}[M_{HPP}^s] \leq \frac{1 - q^s}{q^s \sqrt{\lambda \pi}} \left( \frac{q \Gamma(3/2)}{(1 - q)^{3/2}} - \frac{q \Gamma(3/2)}{(1 - q^2)^{3/2}} - \sum_{k=1}^{s-1} \frac{\Gamma(1/2 + k)q^k(1 - q^{k-1})}{\Gamma(k)} \right).
$$

**Proof.** Using Equation (3.26), with $a = 1$,

$$
\mathbb{E}[M_{s}^{\text{Poisson}}] = \sum_{k=s}^{\infty} \epsilon(k) \cdot \mathbb{P}(\max(x) = k)
$$

$$
= \sum_{k=s}^{\infty} \frac{\Gamma(1/2 + k)}{(\lambda \pi)^{1/2} \Gamma(k)} \cdot \mathbb{P}(\max(x) = k)
$$

$$
\leq \sum_{k=s}^{\infty} \frac{\Gamma(1/2 + k)}{(\lambda \pi)^{1/2} \Gamma(k)} \cdot q^{k-s} (1 - q^s)(1 - q^{k-1})
$$

$$
= \frac{1 - q^s}{q^s \sqrt{\lambda \pi}} \left( \frac{q \Gamma(3/2)}{(1 - q)^{3/2}} - \frac{q \Gamma(3/2)}{(1 - q^2)^{3/2}} - \sum_{k=1}^{s-1} \frac{\Gamma(1/2 + k)q^k(1 - q^{k-1})}{\Gamma(k)} \right).
$$
where Equation (3.29) follows from Observation 3.1 in Section 3.3.4. Equation (3.30) follows from Lemma 3.5 with $a = 1$.

Figure 3.3 shows an upper bound on the maximum expected distance over which a sensor transmits under Greedy for different $s$ and $\lambda$ (intensity of the homogeneous Poisson process according to which the sensors are placed in the plane). This result indicates how fast the batteries of the sensor are expected to be depleted under the Greedy schedule.

So far, we have considered sensor transmission schedules which have a tight time window within which sensors measurements must be retrieved. In particular, we have analyzed the Greedy schedule which requires that in every time slot an innovative sensor measurement is retrieved. In the next section, we will consider more flexible sensor transmission schedule in the sense that the sink has more than $s$ time slots, and therefore, more flexibility, to retrieve $s$ measurements from the sensor network and to make sure that the energy consumed by the transmitting sensors is minimized.

3.5 Energy Consumption - General Time Constraints

In this section we assume a general time constraint $t \geq s$ within which exactly $s$ measurements from $s$ distinct sensors must be retrieved. Every time slot, the sink needs to decide whether to schedule the closest and eligible sensor for transmission or to postpone any transmission for the next slots.

The sink aims to minimize the energy consumption of the sensors. Since the energy consumption of a sensor is an increasing function of the distance between the sensor and the sink, this is equivalent to aiming to schedule sensors as close as possible to the sink. In comparison to the Greedy schedule where, in each time slot, the closest to the sink and eligible sensor will be scheduled for transmission, the sink has now more flexibility when scheduling the sensors.

Figure 3.4 shows the case when $s = 3$ measurements need to be retrieved within $t = 4$ time slots. At time slot 3, the closest and eligible sensor is the 6-th sensor. This sensor is not scheduled and the sink postpones any transmission for the next slot. At time slot 4, the closest and eligible sensor is the 3-rd sensor. The sink must require it to transmit since this is the last slot in which the third measurement can be retrieved. In this example, it was beneficial, from an energy consumption point of view, to postpone any sensor transmission in time slot 3 for the 4 time slot. This is because in the 4-th time slot, the sink observed a closer sensor than the one observed as
Figure 3.2: Total expected energy consumption under Greedy for sensors placed in the plane according to a homogeneous spatial Poisson process with intensity $\lambda$ and path loss exponent of the transmission environment $a$, $s = 5$.

Figure 3.3: Upper bound on the maximum distance over which a sensor transmits under Greedy. The sensors are placed in the plane according to a homogeneous spatial Poisson process with intensity $\lambda$. 
3.5 Energy Consumption - General Time Constraints

closest in slot 3 (thus, the sensor which was required to transmit at slot 4
used less energy).

In Section 3.5.1 we provide a Stochastic Dynamic Programming for-
ulation of the problem of scheduling sensors for transmission under general
time constraints. In Section 3.5.2 we consider a heuristic to schedule the sen-
sors, which requires less computational resources than the formulation in
Section 3.5.1. In Section 3.5.3 we develop an optimal, offline schedule that
assumes that the sink knows ahead of time which sensors will be awake
or asleep in each time slot. We compare numerically the expected energy
consumption of the schedule provide by the Stochastic Dynamic Program-
ming formulation with the heuristic and the off-line schedule considered in
Sections 3.5.2 and 3.5.3.

3.5.1 Stochastic Dynamic Programming

In this section we formulate the problem of scheduling sensors for trans-
mission as a Finite Horizon Stochastic Dynamic Programming (SDP) model
(see, for instance, [81]). We consider a finite horizon model since the time
within which \(s\) measurements need to be retrieved by the sink in order to
retrieve a reliable estimate of an attribute is limited to \(t \geq s\) time slots.

Let \((h, \overline{x})\), with \(|h| = i, 0 \leq i \leq s, h = \{x_1, \ldots, x_i\}, h \subset \mathbb{N}^+\) and
\(\overline{x} \in \mathbb{N}^+, \overline{x} \notin h\), denote the state in which the set \(h\) of sensors that have
transmitted in the previous slots. Thus, \(s - |h|\) measurements still need to be
retrieved and \(\overline{x}\) is the closest to the sink and eligible sensor observed at the
current time slot.

As an example, for \(s = 3\) measurements that need to be retrieved by the
sink, the state \((\{7, 2\}, 5)\) is the situation where 2 measurements have been
retrieved in the previous slots by the sink, i.e. \(h = \{7, 2\}\), the first sensor
that transmitted was the 7–th closest sensor to the sink, i.e. \(x_1 = 7\), the
second sensor transmitting was the second closest sensor, i.e. \(x_2 = 2\), and
the currently observed, eligible and closest sensor is the 5–th closest sensor,
i.e. \(\overline{x} = 5\).

Every time slot, \(\overline{x}\) is either scheduled for transmission or not. When \(\overline{x}\)
is scheduled, a cost \(e(\overline{x})\) is incurred. This is the energy needed by \(\overline{x}\) to trans-
mit to the sink. The SDP model aims to schedule for transmission exactly \(s\)
distinct sensors within \(t\) time slots such that the total energy consumption is
minimal.

Let \(P(\overline{x}|h)\) denote the probability that \(\overline{x}\) is the closest eligible sensor seen
at a time slot, given that the set \(h\) transmitted. Let \(k, k \leq \overline{x} - 1\), denote the
Figure 3.4: Example of a realization of a sensor transmission schedule where \( s = 3 \) measurements from distinct sensors need to be retrieved by the sink within \( t = 4 \) time slots. The sensors are sorted according to their distance to the sink. Every time slot, the closest eligible sensor to the sink may be scheduled for transmission. A black, filled circle indicates an awake sensor. An empty circle indicates an asleep sensor. A rectangle indicates a non-eligible sensor.
number of sensors that are closer to the sink than $\mathcal{F}$ and that have transmitted in the previous slots. Then,

$$P(\mathcal{F}|h) = pq^{\mathcal{F} - 1 - k}.$$  

We consider the following value function, where $1 \leq n \leq t$:

$$F_n(h, y) = \min \left\{ \sum_{\mathcal{F}=1}^{\infty} F_n(h, \mathcal{F})P(\mathcal{F}|h), e(y) + \sum_{\mathcal{F}=1}^{\infty} F_{n+1}(h \cup \mathcal{F}, \mathcal{F})P(\mathcal{F}|h \cup \{y\}) \right\}.$$  

(3.31)

The first argument of the minimum in Equation (3.31) corresponds to not scheduling the closest eligible sensor observed at slot $n + 1$. Thus, $s - |h|$ measurements still need to be retrieved at the sink. The second argument of the minimum in Equation (3.31) corresponds to scheduling the closest eligible sensor for transmission. In this case, we incur a cost $e(y)$, the energy needed by sensor $y$ to transmit to the sink, and we are left with $s - (|h| + 1)$ sensor measurements which still need to be retrieved at the sink.

To ensure that we select exactly $s$ measurements, we define the boundary conditions:

$$\begin{align*}
F_{t+1}(h, y) &= \infty, \forall i \neq s, |h| = i, y \notin h, \\
F_{t+1}(h, y) &= 0, |h| = s, y \notin h.
\end{align*}$$

The boundary conditions above penalize all those schedules that result in the retrieval of less or more than $s$ measurements in $t$ time slots. The value function in Equation (3.31) is a minimization function. Thus, we consider a boundary condition that adds an infinite cost for all those schedules that result in more or less than $s$ measurements retrieved. For all those schedules that result in the retrieval of exactly $s$ measurements, we consider a boundary condition with zero costs.

We use backward induction (see, for instance, [81], Section 4.5) to solve Equation (3.31). Thereby, we obtain $F_1(\emptyset, y), \forall y \in \mathbb{N}^+$, where $(\emptyset, y)$ is the starting state of the system. Then, we determine a minimal expected energy consumption used for the retrieval of a reliable estimate, based on exactly $s$ measurements from $s$ distinct sensors, as:

$$E[P^*_s] = \sum_{y \geq 1} F_1(\emptyset, y)P(y|\emptyset).$$
Simultaneously, we obtain an optimal transmission schedule for the sensors (by keeping track of which sensor, observed as the closest and eligible, is scheduled for transmission at which slot and for which state of the system).

The SDP formulation above has an infinite state space due to the fact that we consider all sensors in the plane. To make the problem computationally tractable, we will truncate the state space, i.e., we will consider the closest $m$ sensors to the sink, where $m$ is a large, but finite number.

### 3.5.2 Heuristic

In this section we consider a simple heuristic to determine which sensors are scheduled for transmission such that a reliable estimate is retrieved from the sensors within a predefined time window of $t$ time slots. This heuristic schedules for transmission, in a time slot, the closest eligible sensor if the expected energy consumption of this sensor is at most the expected energy consumption when all sensors, that have not transmitted in a previous slot, are considered. Below we introduce this heuristic formally.

At a slot $1 \leq n \leq t$, assume that we have retrieved the measurements from a set of sensors $h = \{x_1, \ldots, x_i\}, |h| = i, 0 \leq i < s$ and that we observe $\overline{x} \notin h$, which is the closest awake and eligible sensor. The heuristic selects for transmission $\overline{x}$ if

$$e(\overline{x}) \leq \sum_{u \geq 1 \atop u \notin h} e(u)P(u|h),$$

where $e(u) = \delta_u$. Recall that $P(u|h)$ is the probability that sensor $u$ is the closest, eligible sensor, given that the set $h$ of sensors has transmitted previously. Therefore, $P(u|h) = p(1 - p)^{u-1-k}, k \leq u - 1$, where $k$ is the number of sensors closer to the sink than $u$ and that are in $h$. If $\overline{x}$ is selected, then $h = h \cup \{\overline{x}\}$.

Moreover, if

$$t - n \leq s - |h|,$$

then $\overline{x}$, with $\overline{x} \notin h$, is selected and $h = h \cup \{\overline{x}\}$. This ensures that exactly $s$ measurements from $s$ distinct sensors are retrieved within $t$ time slots, $t \geq s$.

### 3.5.3 Offline Schedule

In this section we consider an offline sensor transmission schedule which assumes that the asleep/awake state of the sensors in every time slot is known ahead of time, i.e. before starting to schedule sensors for transmission. Given this assumption, we determine the best possible schedule that
minimizes the energy consumption of the transmitting sensors while ensuring that sufficient sensor measurements are retrieved at the sink to compute a reliable estimate $\bar{x}$ and that the time constraints are met. We call this schedule Offline. The Offline schedule provides a lower bound for the energy consumption achieved under the SDP schedule.

The Offline schedule corresponds to a minimum weight matching in a bipartite graph that we describe next.

First, we introduce some basic concepts about matching in a bipartite graph. Let $G = (V, E)$ be a bipartite graph, where $V$ is partitioned into two vertex classes, $T$ and $N$, $E \subset T \times N$, i.e., an edge in $E$ has one end point in $T$ and the other end point in $N$.

A matching $M \subset E$ is a collection of edges such that every vertex in $T$ is incident to at most one edge of $M$. The size of a matching $M$ is the number of edges of $M$.

Consider the weight function $w : E \to \mathbb{R}$. Given a weight $w_{i,j}$ for all edges $(i, j) \in E$, the weight of a matching $M$ is $\sum_{(i,j) \in M} w_{i,j}$.

We next show the correspondence between a minimum weight bipartite graph and the Offline sensor transmission schedule.

Let $T = \{1, \ldots, t\}$ correspond to the time slots within which sensor measurements are retrieved and $N \subset \mathbb{N}^+$ consists of the sensors that are awake at least once during the time slots $1, 2, \ldots, t$. An edge $g = \{i, j\}$, with $i \in T$ and $j \in N$, exists if sensor $j$ is awake in time slot $i$.

Let $w(j)$, with $j \in N$, be the weight of node $j$, i.e. the energy used by sensor $j$ to transmit to the sink. Let $w(g) = w(j), g = \{i, j\}, i \in T$ and $j \in N$. Thus, the weight of an edge depends only on the elements in $N$. Clearly, a matching of size $s \leq t$ in the graph $G$ corresponds to a feasible schedule where at each time slot at most one sensor transmits and $s$ sensors transmit
in the time slots \{1, 2, \ldots, t\}. Moreover, no sensor transmits in more than one time slot.

We aim to find a minimum weight matching of size \(s\) on \(G\), which will provide the Offline schedule. Observe that the weight of a matching depends only on the nodes \(j \in N\) in the matching. We proceed in two steps:

i) We compute a minimum weight perfect matching on \(G\). Clearly, the size of this matching is \(t\). We determine such a matching using, for example, the Hungarian algorithm [57].

ii) We determine a matching of size \(s\) by eliminating the \(t - s\) largest weight edges from the matching previously found. We refer to this two step algorithm as the Selection algorithm.

The optimality of the Selection algorithm follows from the next result.

**Lemma 3.6.** A minimum weight matching of size \(s\) on graph \(G\) can be found by eliminating the \(t - s\) largest weight edges from a minimum weight perfect matching on graph \(G\).

**Proof.** Let \(M_{Select}\) be a matching on \(G\) determined by the Selection algorithm. Let \(M_{Opt}\) be an optimal minimum weight matching on \(G\) of size \(s\).

Let \(Y_{Select}\), \(Y_{Select} \subset N\) and \(Y_{Opt}\), \(Y_{Opt} \subset N\) be the set of nodes, which correspond to the sensors selected for transmission, in the optimal solution of \(M_{Select}\) and \(M_{Opt}\), respectively. Note that \(|Y_{Select}| = |Y_{Opt}| = s\).

The proof follows by showing that \(\sum_{j \in Y_{Select}} w(j) = \sum_{i \in Y_{Opt}} w(i)\), i.e., showing that \(M_{Select}\) and \(M_{Opt}\) have the same weight.

Assume that \(Y_{Select} \neq Y_{Opt}\), i.e. \(\sum_{j \in Y_{Select}} w(j) \neq \sum_{i \in Y_{Opt}} w(i)\).

Let \(w(b) = \min_{j \in Y_{Opt} \setminus Y_{Select}} w(j)\) and \(w(u) = \max_{j \in Y_{Select} \setminus Y_{Opt}} w(j)\).

Let the edge \(\{b, t_b\}\) be part of \(M_{Opt}\). Assume that \(t_b\) is not involved in \(M_{Select}\). Then we can lower the weight of \(M_{Select}\) by adding the edge \(\{b, t_b\}\) and removing node \(u\) and the edge incident to it. This contradicts the assumption that \(M_{Select}\) is obtained by the Selection algorithm, which eliminates the \(t - s\) largest weight edges from a minimum weight perfect matching on \(G\) (of size \(t\)). Otherwise, in the Selection algorithm we could have removed all edges incident to node \(u\) as well. Therefore, \(t_b\) must be part of \(M_{Select}\).

Now assume that the edge \(\{t_b, b'\}\) is part of \(M_{Select}\). If \(w(b') > w(b)\), \(b \in N, b' \in N\), then we can remove \(\{t_b, b'\}\) from \(M_{Select}\) and add \(\{t_b, b\}\) to get a lower weight of the matching. This contradicts again the fact that \(M_{Select}\) is obtained by the Selection algorithm. Thus, we may assume that \(w(b') \leq w(b)\).
Suppose that $w(b') < w(b)$, $b \in \mathcal{N}, b' \in \mathcal{N}$. But now we can lower the weight of $M_{Opt}$ by replacing $\{t_b, b\}$ with $\{t_{b'}, b'\}$. This contradicts the optimality of $M_{Opt}$, i.e., optimal minimum weight matching on $G$ of size $s$.

Therefore, $M_{Select}$ and $M_{Opt}$ must have the same weight. \hfill \Box

## 3.6 Numerical results

In this section we analyze numerically the performance of the SDP schedule in Section 3.5.1 with the heuristic considered in Section 3.5.2 and an optimal offline schedule considered in Section 3.5.3.

Since the scheduling formulations in Sections 3.5.1,3.5.2 and 3.5.3 assume an infinite state space, for computational reasons, in this section we assume that for all sensors farther away than the $m$-th sensor, where $m$ is large, $m >> s$, the energy consumption is always $\Phi$, with $\Phi$ large and $e(m) < \Phi$, $e(m) = \delta_m$. Thus, we consider $m+1$ sensors that can be scheduled, where the $(m+1)$th sensor can be viewed as a special sensor that is always awake and eligible for transmission. This could also be viewed as a device or base station that can always provide measurements from the sensors farther away than the $m$-th sensor, but at a higher, fixed cost $\Phi$.

Note that the SDP schedules the $(m+1)$th sensor only in the last slots, when the number of remaining measurements to retrieve is equal to the remaining number of time slots for scheduling, which is when the SDP is forced to schedule a sensor to ensure that, in the end, there are sufficient measurements retrieved.

Figures 3.6 and 3.7 show the expected energy consumption under the SDP schedule, heuristic and the Offline schedule, where the expectation is taken over the randomness of the process according to which the sensors are placed in the plane and over the randomness of the asleep/awake state of the sensors. The energy consumption of sensor $k$ is assumed to be $e(k) = \delta_k$ (for an explanation of the energy function, see Section 3.4).

Figure 3.6 shows the total expected energy consumption for sensors when $s = 2$ measurements need to be retrieved at the sink within $2 \leq t \leq 8$ time slots. As expected, the Offline schedule results in the lowest energy consumption. This is due to the fact that the sink has knowledge about the asleep/awake state of the sensors ahead of time. The SDP and the heuristic schedule sensors in an online fashion, where the lack of knowledge about the asleep/awake state of the sensors leads to larger energy consumptions. The energy consumption under the SDP schedule converges to the energy of the Offline schedule when the time for measurement retrieval increases.
For the case $s = t = 2$, both the SDP schedule and the heuristic result in the Greedy schedule, i.e. every time slot, the closest to the sink and eligible sensor is scheduled for transmission. Even for $t = s = 2$, the Offline schedule results in a lower energy consumption than the SDP schedule and the heuristic.

To illustrate how the energy consumption improves under the Offline schedule for $t = s = 2$, consider the following example of a realization of Offline and SDP schedule. Assume that in the first time slot, i.e., $t = 1$, the closest and awake 2 sensors are the 3-rd and the 4-th closest sensors to the sink. These sensors require an energy consumption $\delta_3^a$ and $\delta_4^a$, respectively, with $\delta_3^a < \delta_4^a$. Now assume that in the second time slot, i.e., $t = 2$, the closest and awake 2 sensors are the 3-rd and the 10-th closest sensor to the sink. These sensors require an energy consumption $\delta_3^a$ and $\delta_{10}^a$, respectively, with $\delta_3^a < \delta_4^a < \delta_{10}^a$. The SDP schedules online sensors for transmission and because $t = s$, it is forced to select each time slot the closest, awake and eligible sensor. Thus, SDP schedules the 3-rd closest sensor in the first time slot and the 10-th closest sensor in the second time slot, with a total energy consump-
3.6 Numerical results

Figure 3.7: Expected energy consumption - Sensors randomly placed in the plane according to a homogeneous spatial Poisson process with intensity $\lambda = 0.1$, $s = 4, p = 0.5, a = 2$.  

The Offline schedule, however, knows ahead of time which sensors are awake and during which time slots, so it will schedule the 4-th closest sensor in the first slot and the 3-rd closest sensor in the second slot, with a total energy consumption of $\delta_4^{a} + \delta_3^{a}$, which, clearly, is lower than the total energy consumption achieved under the SDP schedule.

For small $t$, Figure 3.6 shows that the SDP schedule and the heuristic result in similar energy consumptions. This because the finite horizon over which the SDP optimizes the energy consumption of the transmitting sensors is limited.

As $t$ increases, Figure 3.6 shows that the energy consumption under the SDP schedule decreases further. This is because the SDP schedule has more flexibility in scheduling the sensors as the time allocated for measurement retrieval increases. The heuristics improves on the energy consumption up to a certain $t$, after which increasing the time for measurement retrieval does not reduce further the energy consumption. This is because the heuristic does not take into account the residual time for measurement retrieval (with
the exception of the last time slots when sufficient measurements are not retrieved and the heuristic is forced to schedule sensors). Instead, the heuristic schedules sensors for transmission based on their energy consumption, which is preferably below the average energy consumption of sensors that are eligible for scheduling.

Figure 3.7 shows the total expected energy consumption for sensors when \( s = 4 \) measurements need to be retrieved at the sink within \( 4 \leq t \leq 8 \) time slots. Again, as expected, the Offline schedule results in the lowest energy consumption. The SDP schedule results in a lower energy consumption as \( t \) increases. In comparison to the SDP schedule, the heuristic leads to higher energy consumption (in this example, the heuristic leads to approximately 10% higher energy consumption than the SDP schedule).

3.7 Conclusions

In this chapter, we considered the problem of retrieving a reliable estimate of an attribute from a wireless sensor network within a predefined time window and with minimal energy consumption for the sensors.

Firstly, we considered sensor transmission schedules with tight time constraints. In particular, we analyzed a Greedy schedule which retrieves \( s \) measurements in \( s \) time slots from the closest, eligible sensor in each of these slots. We determined a closed-form expression for the energy consumption of sensors under a Greedy schedule, when \( s \) measurements need to be retrieved by the sink within \( s \) time slots. We also determined for this Greedy schedule the maximum expected distance over which a sensor transmits to the sink. This results are general in the sense that they can be applied for any type of energy consumption function and for any distribution of the sensors in the plane. To illustrate this, we applied our results for the case when sensors placed in the plane according to a homogeneous spatial Poisson process.

Secondly, we considered sensor transmission schedules under general time constraints, that minimize sensor energy consumption. In this case, we provided a stochastic dynamic programming model (SDP) to optimally schedule sensors with respect to the sensor energy consumption. We also developed one heuristic and an offline, optimal schedule, which assumes that the asleep/awake state of the sensors is known ahead of time, to schedule sensors for transmission such that a reliable estimate is retrieved within a predefined time. We compared numerically the performance of the SDP schedule with the heuristic and the Offline schedule and showed numeri-
cally that the SDP schedule converges to the Offline schedule as the time allocated for measurement retrieval increases. We also showed that the heuristic performs close to the SDP schedule in terms of energy consumption.

Our results provide insight into the impact of the time allocated for measurement retrieval, and number of measurements to be retrieved such that the reliability of an estimate of an attribute is ensured with minimal energy consumption for the sensors.
3.8 Appendix

3.8.1 Proof of Lemma 3.2

Proof.

\[ \sum_{\tau(x) \in O_s} \left( \sum_{i=1}^{s} \epsilon(\tau(x)_i) \right) p_x = \frac{1}{s!} \sum_{\tau(x) \in D_s} \left( \sum_{i=1}^{s} \epsilon(\tau(x)_i) \right) p_x \]  \hspace{1cm} (3.32)

\[ = \frac{1}{s!} \sum_{\tau(x) \in D_s} \left( \sum_{i=1}^{s} \epsilon(\tau(x)_1) \right) p_x \]  \hspace{1cm} (3.33)

\[ = \frac{s}{s!} \sum_{\tau(x) \in D_s} \epsilon(\tau(x)_1) p_x, \]  \hspace{1cm} (3.34)

where (3.32) follows from considering all permutations of the set \( x, |x| = s \). and (3.33) follows from the observation that

\[ \sum_{\tau(x) \in D_s} \epsilon(\tau(x)_i) p_x = \sum_{\tau(x) \in D_s} \epsilon(\tau(x)_j) p_x, \]

for \( i \neq j \) and \( 1 \leq i \leq s, 1 \leq j \leq s \). \( \square \)

3.8.2 Proof of Lemma 3.3

We first state the following lemma, used in the proof of Lemma 3.3.

Lemma 3.7. For \( s \geq 1 \),

\[ \sum_{\tau(x) \in D_s} \prod_{i=1}^{s} q^{\tau(x)_i} = s! \frac{q^{s(s+1)/2}}{\prod_{i=1}^{s} (1 - q^i)}. \]

Proof.

\[ \sum_{\tau(x) \in D_s} \prod_{i=1}^{s} q^{\tau(x)_i} = s! \sum_{\tau(x) \in O_s} \prod_{i=1}^{s} q^{\tau(x)_i}, \]  \hspace{1cm} (3.35)

\[ = s! \frac{q}{1-q} \left( \sum_{\tau(x) \in O_{s-1}} \left( \prod_{i=1}^{s-2} q^{\tau(x)_i} \right) q^{2\tau(x)_{s-1}} \right) \]  \hspace{1cm} (3.36)

\[ = s! \frac{q^2}{1-q} \left( \sum_{\tau(x) \in O_{s-2}} \left( \prod_{i=1}^{s-3} q^{\tau(x)_i} \right) q^{3\tau(x)_{s-2}} \right). \]  \hspace{1cm} (3.37)
where (3.35) follows from considering only the permutation of set $x$ which has all components in increasing order; (3.36) follows after computing the sum over $\tau(x)_s$ and using the fact that

$$\sum_{\tau(x)_s > \tau(x)_{s-1}} \infty \frac{q^{\tau(x)_s}}{1 - q} = \frac{q}{1 - q} q^{\tau(x)_{s-1}}.$$ 

Equation (3.37) follows after computing the sum over $\tau(x)_{s-1}$ and using the fact that

$$\sum_{\tau(x)_{s-1} > \tau(x)_{s-2}} \infty \frac{q^{2\tau(x)_{s-1}}}{1 - q^2} q^{2\tau(x)_{s-2}} = \frac{q^2}{1 - q^2} q^{2\tau(x)_{s-2}}.$$ 

Continuing the same reasoning and using the fact that

$$\sum_{j=k} \infty \frac{q^a}{1 - q^a} q^{ak},$$

the result follows.

We can now state the proof of Lemma 3.3:

**Proof.** The proof is constructed using induction on the size of the set $x \subset \mathbb{N}^+$ and Lemma 3.7. First, we define:

$$W_s = \sum_{\tau(x) \in D_s} \epsilon(\tau(x)_1) \prod_{i=1}^{s} q^{\tau(x)_i}.$$ 

Let $|x| = 2$. Then,

$$W_2 = \sum_{\tau(x) \in D_2} \epsilon(\tau(x)_1) \cdot q^{\tau(x)_1+\tau(x)_2}$$

$$= \sum_{\tau(x)_1 = 1}^{\infty} \epsilon(\tau(x)_1) \cdot q^{\tau(x)_1} \left( \sum_{\tau(x)_2 = 1}^{\infty} q^{\tau(x)_2} - q^{\tau(x)_1} \right)$$

$$= \sum_{\tau(x)_1 = 1}^{\infty} \epsilon(\tau(x)_1) \cdot q^{\tau(x)_1} \left( \frac{q}{1 - q} - q^{\tau(x)_1} \right).$$
We now assume that Lemma 3.3 holds for $|x| = s, s > 2$, so

$$W_s = (s - 1)! \sum_{\tau(x)_1 = 1}^{\infty} \epsilon(\tau(x)_1) \cdot q^{\tau(x)_1} \sum_{j=1}^{s} \frac{(-1)^{j+1} q^{(s-j)(s-j+1)/2} q^{(j-1)\tau(x)_1}}{\prod_{i=1}^{s-j} (1 - q^i)}.$$ 

We now consider the case $|x| = s + 1$. Then,

$$W_{s+1} = \sum_{\tau(x) \in D_{s+1}} \epsilon(\tau(x)_1) \prod_{i=1}^{s+1} q^{\tau(x)_i}$$ (3.38)

$$= \sum_{\tau(x) \in D_{s+1}} \prod_{i=2}^{s+1} q^{\tau(x)_i} \left( \sum_{\tau(x)_{s+1} = 1}^{\infty} \epsilon(\tau(x)_{s+1}) \cdot q^{\tau(x)_{s+1}} - \sum_{i=2}^{s+1} \epsilon(\tau(x)_i) \cdot q^{\tau(x)_i} \right)$$ (3.39)

$$= \sum_{\tau(x) \in D_{s}} \prod_{i=1}^{s} q^{\tau(x)_i} \left( \sum_{\tau(x)_{s+1} = 1}^{\infty} \epsilon(\tau(x)_{s+1}) \cdot q^{\tau(x)_{s+1}} \right)$$ (3.40)

$$- \sum_{\tau(x) \in D_{s}} \prod_{i=1}^{s} q^{\tau(x)_i} \left( \sum_{i=1}^{s} \epsilon(\tau(x)_i) \cdot q^{\tau(x)_i} \right)$$ (3.41)

$$= \sum_{\tau(x) \in D_{s}} \prod_{i=1}^{s} q^{\tau(x)_i} \left( \sum_{\tau(x)_{s+1} = 1}^{\infty} \epsilon(\tau(x)_{s+1}) \cdot q^{\tau(x)_{s+1}} \right)$$ (3.42)

$$- s \sum_{\tau(x) \in D_{s}} \epsilon(\tau(x)_1) \cdot q^{2\tau(x)_1} \prod_{i=2}^{s} q^{\tau(x)_i}$$ (3.43)

$$= s! \frac{q^{s(s+1)/2}}{\prod_{i=1}^{s} (1 - q^i)} \left( \sum_{\tau(x)_{s+1} = 1}^{\infty} \epsilon(\tau(x)_{s+1}) \cdot q^{\tau(x)_{s+1}} \right)$$
Equation (3.39) follows from allowing the first element of the permutation \( \tau(x) \), \(|x| = s + 1 \) to take all values in \( \mathbb{N}^+ \) except all those values of \( \tau(x)_i \), \( 2 \leq i \leq s + 1 \). Equation (3.41) follows from re-denoting, without loss of generality, \( \tau(x)_1 = \tau(x)_{s+1} \) and \( \tau(x)_{s+1} = \tau(x)_1 \). Element \( \tau(x)_{s+1} \) takes now values in \( N^+ \), independently of the other elements \( \tau(x)_1, \ldots, \tau(x)_s \). Equation (3.42)
follows from the symmetry in the elements $\tau(x)_1, \ldots, \tau(x)_s \in D_s$. Equation (3.44) follows from Lemma 3.7. Equation (3.45) follows from re-denoting, without loss of generality, $\tau(x)_{s+1} = \tau(x)_1$. Equation (3.47) follows from substituting $W_s$, according to the induction step. This concludes the proof by induction.

3.8.3 Proof of Lemma 3.4

Proof. Let $k \geq s \geq 1$. Let $x \subset \{1, 2, \ldots, k-1\}$, with $|x| = s - 1$ and $z = \{1, 2, \ldots, k-1\} \setminus x$.

The proof is constructed using induction on the number of elements of $z$ and on $k$. First, we denote:

$$Z(k-s, k-1) = \sum_{\tau(z) \in D_{k-s}(k-1)} \prod_{i=1}^{k-s} q^{-\tau(z)_i}.$$  

The first argument of $Z(k-1, k-s)$ indicates the size of $\tau(z)$. The second argument indicates the maximal value that an element of $\tau(z)$ can take. Clearly, $k-s \leq k-1$.

Now, for $n \geq 1$ and $k-s \geq 1$

$$Z(1, 1) = \sum_{z_1=1}^{1} q^{-z_1} = q^{-1};$$

$$Z(1, n) = \sum_{z_1=1}^{n} q^{-z_1} = q^{-n} \frac{1 - q^n}{1 - q};$$

$$Z(k-s, k-s) = \sum_{z \in D_{k-s}(k-s)} \prod_{i=1}^{k-s} q^{-\tau(z)_i} = (k-s)!q^{-\frac{(k-s)(k-s+1)}{2}}. \tag{3.50}$$

Equation (3.50) follows from the fact that each $\tau(z)_i, 1 \leq i \leq k-s$ can have a value at most $k-s$. Then the set $\{\tau(z)_1, \ldots, \tau(z)_{k-s}\}$ takes values according to one of the $(k-s)!$ permutations of the set $\{1, 2, \ldots, k-s\}$.

We assume that for $n \geq k-s$ and $k \geq s \geq 1$:

$$Z(k-s, n) = (k-s)!q^{-\frac{(k-s)(k-s+1)}{2}} \prod_{i=1}^{k-s} \frac{1 - q^{n+i-1}}{1 - q^i}. $$
Now, for $n \geq k - s$ and $k \geq s \geq 1$:

$$Z(k - s, n + 1) = \sum_{\tau(z) \in \mathcal{D}_{k-s}(n+1)} \prod_{i=1}^{k-s} q^{-\tau(z)_i}$$

$$= (k - s)q^{-(n+k-s)}A(k - s - 1, n + 1) + A(k - s, n)$$

$$= (k - s)!q^{-(n+k-s)}q^{-(k-s-1)(n+1) - \frac{(k-s-1)(k-s-2)}{2}}$$

$$\cdot \prod_{i=1}^{k-s-1} \frac{1 - q^{n+k-s-1}}{1 - q^i}$$

$$+ (k - s)!q^{-(k-s)n - \frac{(k-s)(k-s-1)}{2}} \prod_{i=1}^{k-s-1} \frac{1 - q^{n+i-1}}{1 - q^i} \frac{1}{1 - q^{k-s}}$$

$$= (k - s)!q^{-(k-s)(n+1) - \frac{(k-s)(k-s-1)}{2}} \prod_{i=1}^{k-s-1} \frac{1 - q^{n+i}}{1 - q^i} \frac{1}{1 - q^{k-s}}$$

$$\cdot \left(1 - q^{k-s} + q^{k-s}(1 - q^n)\right)$$

$$= (k - s)!q^{-(k-s)(n+1) - \frac{(k-s)(k-s-1)}{2}} \prod_{i=1}^{k-s} \frac{(1 - q^{n+i})}{(1 - q^i)}.$$
\[= (k - s + 1)!q^{-(k-s+1)n-(k-s)(k-s+1)/2} \prod_{i=1}^{k-s+1} \frac{(1 - q^{n+i-1})}{(1 - q^i)}.\]

This concludes the proof by induction. \qed
CHAPTER 4

Data Retrieval Time for Energy Harvesting Sensor Networks

4.1 Introduction

Managing energy consumption is an important component of wireless sensor network design, as it can lead to increased throughput and network lifetime. Recent technological advances have enabled sensor to harvest energy from the environment and to use this energy to recharge their batteries. Several technologies have been shown to be possible for energy harvesting such as solar energy (Figure 4.1), radio frequency, vibrations, thermoelectric (see, for instance, [79] for various examples of energy harvesting technologies).

In the case of battery-limited sensor networks, one of the main design goals is to minimize the sensor energy consumption so that the lifetime of the sensor network is extended. In contrast, energy harvesting techniques enable sensors to recharge their batteries over time, having the potential to extend the lifetime of the sensor network and to improve the overall performance of the network. However, it is often the case that the amount of en-

Figure 4.1: Example of a heliomote solar harvesting sensor in a weather-proof case [60]
ergy available for harvesting varies over time in a non-deterministic manner. For instance, solar energy varies throughout the day and the intensity of the direct sunlight cannot be controlled. To enhance energy availability, sensor are often equipped with batteries where energy is stored for later use. The storage capacity of the batteries, however, is often limited. It is, therefore, important to develop mechanisms to match the energy generation profile of the harvesting sensors and their energy storage capabilities with the energy consumption of the sensors. In particular, since sensors consume most of their energy to transmit their measurements [3], it is important to jointly consider sensor transmissions and the process of energy harvesting.

Energy harvesting for wireless communications has resulted in a significant number of research publications in the last decade [91]. Considerable attention has been given to the coordination between sensor transmission scheduling, which is an energy-dependent process, and energy harvesting, which depends on the availability of the energy source. Various metrics for data transmission in energy harvesting communication systems have been considered. In [10, 103, 104] the minimization of the time to transmit a fixed number of bits using a AWGN broadcast channel, where the transmitter harvests energy and has a finite-capacity rechargeable battery, is considered. In contrast to the model proposed in this chapter, the energy arrival process is assumed to be known in advance, in an off-line manner. The authors optimize the transmission rate or the transmission power based on the energy available at the transmitter and on the amount of data that needs to be transmitted such that the transmission time is minimized. Structural properties of the transmission policies are derived. Similar to the model proposed in this chapter, in [87, 92] the process of energy harvesting is stochastic. Transmission policies that maximize the rate of data transmission to minimize the mean delay of data transmission are derived. In [92], the average delay of data packets arriving according to a Poisson process at a transmitter which harvests energy, is derived. In comparison, in this chapter we compute the expected delay of a set of sensor measurements to be transmitted by distinct sensors, each harvesting energy according to a Poisson process.

The problem of maximizing the amount of data transmitted up to a certain point is considered in [29, 46, 77, 94]. In [29] a general framework is provided to maximize the amount of transmitted data by a given deadline when the arrival process of energy is known in an off-line manner, at the transmitter and the battery of the transmitter is limited or suffers from energy leakage. In [46,77,94] dynamic programming is employed to determine an optimal energy allocation policy over a finite horizon so that the number
of transmitted bits is maximized. In [59] optimal transmission policies are derived to specify whether to transmit incoming data packets or to drop them based on a value attached to each packet and on the energy available at the transmitter.

Optimal transmission policies with energy harvesting nodes that transmit using fading wireless channels are considered in [46, 56, 77]. In [56], the probability of successful reception of data packets and the energy cost per transmitted packet are determined for energy harvesting devices that broadcast using non-perfect transmission channels. The authors propose an erasure-based broadcast scheme to guarantee reliable transmissions.

This chapter considers the problem of retrieving a fixed number of sensor measurements over an attribute from distinct wireless sensors that harvest energy from the environment. Energy is harvested by each sensor according to a Poisson process, independently of the other sensors. The fact that energy is harvested at random points in time reflects the stochastic nature of the availability of energy to be harvested. We further assume that the batteries have limited storage capacity. When the battery of a sensor reaches the maximal storage capacity, additional harvested energy is discarded. Provided that they have energy, the sensors broadcast measurements at an exponential rate and independently of each other. Clients arrive at the network according to some Poisson process and are interested in retrieving several measurements over an attribute monitored by the sensors. Based on the retrieved sensor measurements, each client computes an estimate of the attribute. We impose that measurements are retrieved from distinct sensors to avoid the situation where the same measurement, from the same sensor, is retrieved multiple times, which could lead to a biased estimate. We determine the probability distribution of the retrieval time of a reliable estimate of an attribute monitored by the sensor network. Consequently, we provide a closed-form expression for the expected retrieval time of a reliable estimate. We also analyze the retrieval time of a reliable estimate when the capacity of the battery or the rate at which energy is harvested is asymptotically large. These results show the impact of the energy availability at the sensors, as well as the energy storage capabilities of the sensors, on the time required to retrieve a reliable estimate of an attribute from the sensor network.

The remainder of this chapter is organized as follows. In Section 4.2 we formulate the model and the problem statement. In Section 4.3 we determine the distribution of the time for a client to retrieve a reliable estimate of an attribute from the sensor network. We also determine the retrieval time of a reliable estimate when the rate at which energy is harvested and the
maximum capacity of the sensor batteries is asymptotically large. In Section 4.4 we compute numerically the retrieval time of a reliable estimate under various assumptions regarding the size of the sensor network, the energy harvesting rate and the maximum capacity of the sensor batteries. In Section 4.5 we discuss the results and provide conclusions.

4.2 Model and Problem Statement

We consider a network of \( N \) wireless sensors, each having a noisy measurement on an attribute \( \theta \). The measurements are subject to independent and identically distributed additive Gaussian noise with variance \( \sigma^2 \), i.e. \( X_i \sim N(\theta, \sigma^2) \).

Clients arrive at the sensor network according to a Poisson process with rate \( \lambda_a \). Each client is interested in acquiring from the network a sufficiently large set of sensor measurements based on which they can compute a reliable estimate \( \bar{X} \) of \( \theta \).

The following model for the reliability of an estimate has been introduced in Chapters 2 and 3. We will restate it here for completeness. We consider an estimate to be reliable if the variance of the estimate \( \bar{X} \) at each client is below a targeted threshold \( H \). Since

\[
\text{Var}(\bar{X}) = \text{Var}\left(\frac{1}{s} \sum_{i=1}^{s} X_i\right) = \frac{1}{s^2} \sum_{i=1}^{s} \text{Var}(X_i) = \frac{\sigma^2}{s} < H,
\]

we consider \( s = \lceil \frac{\sigma^2}{H} \rceil \). Therefore, any set of \( s \) measurements from distinct sensors is sufficient for the client to retrieve a reliable estimate of \( \theta \). As there are \( N \) sensors which can provide \( N \) distinct measurements, we also assume that \( s \leq N \).

Sensor \( i, 1 \leq i \leq N \), has \( b_i \) units of energy, with \( 0 \leq b_i \leq B \). Each sensor harvests one unit of energy at an exponential rate \( \lambda_e \), independently of the other sensors. The energy is harvested from the environment. If \( b_i = B \) and new energy is harvested, then this harvested energy is discarded.

We further assume that the sensor network broadcasts a measurement at an exponential rate \( \mu \). This rate is shared uniformly by the \( N \) sensors in the network. Therefore, at an exponential rate \( \mu/N \), independently of the other sensors, a random sensor broadcasts its measurement, provided it has sufficient energy.

Upon a broadcast, the energy of the broadcasting sensor decreases by 1 unit (see also Figure 4.2).
4.3 Analysis

Figure 4.2: Birth-and-Death model for a single sensor that broadcasts, provided it has energy, at an exponential rate $\mu/N$ and harvests energy from the environment at an exponential rate $\lambda_e$.

A measurement is retrieved by a client only if it is the first time this sensor broadcasts its measurement to this client. We call such a measurement to be innovative for this client. Therefore, a client does not retrieve a measurement from the same sensor multiple times. Moreover, we assume that clients do not drop innovative measurements.

We are interested in the time, denoted by $W_s$, for a client to retrieve a reliable estimate from the sensor network.

We end this section with some notation that will be useful for working with phase-type distributions. Let $e_k$ be a $k \times 1$ matrix with all unit entries. Let $I_k$ denote the $k \times k$ identity matrix. For $n \times n$ matrix $M_1$ and $m \times m$ matrix $M_2$, let $M_1 \otimes M_2$ denote the Kronecker product of matrices $M_1$ and $M_2$ and let $M_1 \oplus M_2$ denote the Kronecker sum, i.e.

$$M_1 \oplus M_2 = M_1 \otimes I_m + I_n \otimes M_2.$$ 

Finally, let $M^\otimes n$ and $M^\oplus n$ denote the $n$-fold Kronecker product and the $n$-fold Kronecker sum with itself, respectively.

4.3 Analysis

In this section we first determine the distribution of the time for a single sensor to broadcast, given that the system is in steady-state. We show that this is a phase-type distribution. Using these results, we next determine the distribution of the time for a random client to retrieve $s$ measurements from distinct sensors. Lastly, we compute the retrieval time of a reliable estimate $\bar{X}$ of $\theta$ for asymptotically large $B$, the maximum capacity of the sensor batteries, $\lambda_e$, the rate at which sensors harvest energy from the environment, and $N$, the size of the sensor network.
4. DATA RETRIEVAL TIME FOR ENERGY HARVESTING SENSORS

4.3.1 A Single Sensor

First, we consider the steady-state probability that a random sensor has $i$ units of energy, $0 \leq i \leq B$, denoted by $\nu(i)$.

The evolution of the units of energy available at a sensor follows a Birth-and-Death model and a finite state space $\{0, 1, \ldots, B\}$ with births at rate $\lambda_e$ and deaths at rate $\mu/N$ (see Figure 4.2). The steady-state distribution of such a model is well-known in literature (see, for instance, [12]) and is, therefore, stated without proof below.

**Lemma 4.1.** The steady-state probability for an arbitrary sensor to have $i$ units of energy, $0 \leq i \leq B$, is:

$$
\nu(i) = \nu_0 \left( \frac{\lambda_e N}{\mu} \right)^i,
$$

(4.1)

where $\nu_0 = (\lambda_e N/\mu - 1)/((\lambda_e N/\mu)^{B+1} - 1)$, if $\lambda_e \neq \mu/N$ and $\nu_0 = 1/(B + 1)$ otherwise.

Note that in the above theorem, $\nu_0$ is the probability that the battery of an arbitrary sensor is depleted.

Next, we consider $W$, the time until an arbitrary sensor broadcasts, given that the system is in steady-state. Based on $W$, we compute the time for a client to retrieve a reliable estimate by assuming that, upon arrival, this client observes the energy level of the sensors in steady-state. This is valid since the sensors operate independently of the arrivals of the clients. Moreover, since clients arrive according to a Poisson process, they see the system in steady-state (PASTA).

Since the evolution of the energy available at an arbitrary sensor follows a continuous-time Markov process, the distribution of $W$ can be modeled as a phase-type distribution as follows. Consider a continuous-time Markov chain with $B + 2$ states. State $0 \leq i \leq B$ are transient states and correspond to a sensor having $i$ units of energy. The $(B + 2)$–th state is an absorbing state. This state is reached when the sensor broadcasts a measurement. At an exponential rate $\lambda_e$, a jump occurs from state $i$ to state $i + 1$, $0 \leq i < B$. This corresponds to the sensor harvesting an additional unit of energy. At an exponential rate $\mu/N$, a transition occurs from state $1 \leq i \leq B$ to the absorbing state. This corresponds to a sensor broadcast. Let the initial distribution over the transient states be $\nu$. Then, the time until absorption is $W$, as desired.

Before giving a more formal description of the phase-type representation of $W$, we make a simplification by observing that in the above description,
the states 1 to $B$ can be aggregated into a single transient state, which we will denote by 1. There is a transition from state 0 to this aggregated state 1 at rate $\lambda_e$ and there is a single outgoing transition from this aggregated state 1 to the absorbing state at rate $\mu/N$. Below, we will give the formal representation of this phase-type distribution as $(a, T)$ and specify the vector $a$ and the matrix $T$. Given this representation as a phase-type distribution, we immediately obtain $\mathbb{P}(W \leq t) = 1 - a e^{T x} e_2$. In this case, however, since $T$ has such a simple structure we can also obtain the distribution function in an explicit form. This yields the following result.

**Lemma 4.2.** The distribution of $W$ is phase-type $(a, T)$, where

$$a = \begin{bmatrix} \nu_0 \\ 1 - \nu_0 \end{bmatrix}, \quad T = \begin{bmatrix} -\lambda & \lambda \\ 0 & -\mu/N \end{bmatrix}. \quad (4.2)$$

The distribution function of $W$ can be expressed as

$$\mathbb{P}(W \leq t) = 1 - e^{-\mu t/N} + \frac{\mu}{N} - \lambda_e e^{-\lambda_e t} \nu(0) \left( e^{-\mu t/N} - e^{-\lambda_e t} \right). \quad (4.3)$$

**Proof.** The phase-type characterization follows from the discussion above the lemma. The expression of the distribution function is obtained by observing that, given that we are in state 0, which happens with probability $\nu(0)$, the distribution of $W$ is given by the sum of two exponentially distributed random variables with parameters $\mu/N$ and $\lambda_e$. Given that we are in the aggregated state 1, which happens with probability $1 - \nu(0)$, the distribution of $W$ is given by an exponentially distributed random variable with parameter $\mu/N$. Therefore,

$$\mathbb{P}(W \leq t) = \left( 1 - \frac{\mu}{N} - \lambda_e e^{-\lambda_e t} + \frac{\mu}{N} - \lambda_e e^{-\mu t/N} \right) \nu(0) + (1 - e^{-\mu t/N})(1 - \nu(0)),$$

The result follows directly from the above expression. \qed

### 4.3.2 Retrieving a Reliable Estimate

In this section, we determine the distribution of the time for a random client to retrieve $s$ measurements from distinct sensors.

**Lemma 4.3.** The distribution of the time until a client receives $s$ measurements from distinct sensors is:

$$\mathbb{P}(W_s \leq t) = 1 - \sum_{j=0}^{s-1} \binom{N}{j} \left( \sum_{k=0}^{j} \binom{j}{k} (-1)^{j-k} a^\otimes(N-K) \exp \left( t T^\otimes(N-K) \right) e_{2N-k} \right).$$
Recall that a client leaves the system as soon as he retrieves exactly $s$ measurements. Thus, we need to compute the distribution of the time between the moment a client arrives at the network and the moment when the $s$-th broadcast occurs, all $s$ broadcasts from distinct sensors. This can be seen as the distribution of the $s$-th order statistic of $N$ phase-type distributed random variables with representation $(a, T)$, as introduced above. The distribution of the $s$-th order statistic is (see, for instance, [2]), for $N$ variables,

$P(W_s \leq t) = \sum_{j=s}^{N} \binom{N}{j} P(W \leq t)^j (P(W > t)^{N-j}).$ (4.4)

Starting from the above expression we derive

$P(W_s \leq t) = 1 - \sum_{j=0}^{s-1} \binom{N}{j} (1 - P(W > t))^j (P(W > t)^{N-j})$

$= 1 - \sum_{j=0}^{s-1} \binom{N}{j} \left( \sum_{k=0}^{j} \binom{j}{k} (-1)^{j-k} P(W > t)^{j-k} + 1 \right) (P(W > t)^{N-j})$ (4.5)

$= 1 - \sum_{j=0}^{s-1} \binom{N}{j} \left( \sum_{k=0}^{j} \binom{j}{k} (-1)^{j-k} P(W > t)^{N-k} \right),$ (4.6)

where in (4.5) we expanded the polynomial $(1 - P(W > t))^j$.

Now, observe that the distribution of $P(W > t)^{N-k}$ in (4.6) is:

$P(W > t)^{N-k} = P(\min\{Y_1, Y_2, \ldots, Y_{N-k}\} > t),$ (4.7)

where the $Y_i, 1 \leq i \leq N - k$ are i.i.d. phase-type distributed random variables with representation $(a, T)$. Therefore, $P(W > t)^{N-k}$ is the first order statistic of a phase-type distributed random variable for which it is well known (see, for instance, [74]) that it is phase-type distributed with representation $(a \otimes (N-k), T \oplus (N-k))$. The result follows directly by inserting the distribution function of this phase-type distribution into (4.6). \hfill \Box

The result above is general in the sense that it does not depend on the specific representation of the phase-type distributed random variable $W$. 

\begin{proof}

\end{proof}
We are next interested in determining $\mathbb{E}[W_s]$, the expected time for a client to retrieve exactly $s$ measurements. In principle, $\mathbb{E}[W_s]$ can be obtained directly from Lemma 4.3. However, the moments of order statistics of phase-type distributed random variables are known in the literature [113]. Therefore, we will resort to the results from [113]. Let $m^k_s$ denote the $k$-th moment of the $s$-th order statistic of $N$ phase-type distributed random variables with representation $(a, T)$. Then, from [113] we have the following result.

**Theorem 4.1.** [113, Thm 4.1]

$$m^k_s = m^k_{s-1} + \sum_{j=1}^{s} (-1)^{j-1} \binom{N-s+j}{j-1} L^{(k)}_{N-s+j},$$

where $L^{(k)}_j = \binom{N}{j} (-1)^k k! (a^j) (T^j)^{-k} e_{2j}, 1 \leq j \leq s$, and $m^k_0 = 0$.

Taking $k = 1$, $\mathbb{E}[W_s]$ can be computed from Theorem 4.1. Note, however, that straightforward computation of moments based on Theorem 4.1 involves the matrices $T^{\otimes j}$, where $j$ takes values up to $N$. The dimension of $T^{\otimes N}$ is $2N \times 2N$. Therefore, the complexity of these computations is exponentially increasing in $N$. Since we are interested in the behaviour of the system for larger values of $N$, we will derive in the next result an expression for $\mathbb{E}[W_s]$ that has at most polynomial complexity in all model parameters.

**Theorem 4.2.** The expected time, $\mathbb{E}[W_s]$, for a client to retrieve $s$ measurements from distinct sensors is:

$$\sum_{j=0}^{s-1} \binom{N}{j} \sum_{k=0}^{j} \binom{j}{k} (-1)^{j-k} \sum_{v=0}^{N-k} \binom{N-k}{v} \omega^v (1-\omega)^{N-k-v} \frac{1}{\lambda_e (N-k-v) + \frac{\mu}{N} v},$$

where $\omega = 1 - \nu(0) \frac{\mu}{N - \lambda_e}$ and $\nu(0)$ is the steady-state probability of an arbitrary sensor having no energy.

**Proof.** The expected retrieval time for $s$ measurements from distinct sensors
can be expressed using Theorem 4.3 and Lemma 4.2 as follows.

\[
\mathbb{E}[W_s] = \int_0^\infty \mathbb{P}(W_s > t) dt = \sum_{j=0}^{s-1} \binom{N}{j} \int_0^\infty \left( \sum_{k=0}^j \binom{j}{k} (-1)^{N-k} (1 - \mathbb{P}(W \leq t))^{N-k} \right) dt \tag{4.8}
\]

\[
= \sum_{j=0}^{s-1} \binom{N}{j} \sum_{k=0}^j \binom{j}{k} (-1)^{N-k} \int_0^\infty \left( e^{-\mu t} \omega + e^{-\lambda t} (1 - \omega)^N \right)^{N-k} dt \tag{4.9}
\]

\[
= \sum_{j=0}^{s-1} \binom{N}{j} \sum_{k=0}^j \binom{j}{k} (-1)^{N-k} \int_0^\infty \sum_{v=0}^N \binom{N-k}{v} \left( e^{-\mu t} \omega \right)^v \left( e^{-\lambda t} (1 - \omega)^N \right)^{N-k-v} dt
\]

\[
= \sum_{j=0}^{s-1} \binom{N}{j} \sum_{k=0}^j \binom{j}{k} (-1)^{N-k} \sum_{v=0}^N \binom{N-k}{v} \omega^v (1 - \omega)^{N-k-v} \cdot \int_0^\infty \left( e^{-\mu t} \right)^v \left( e^{-\lambda t} \right)^{N-k-v} dt
\]

\[
= \sum_{j=0}^{s-1} \binom{N}{j} \sum_{k=0}^j \binom{j}{k} (-1)^{N-k} \sum_{v=0}^N \binom{N-k}{v} \omega^v (1 - \omega)^{N-k-v} \frac{\omega^v (1 - \omega)^{N-k-v}}{\lambda e (N-k-v) + \frac{\mu}{N}} + \mu \frac{N}{N} \tag{4.10}
\]

where (4.8) follows from the derivations in (4.6) and (4.9) follows from Lemma 4.2, where we denoted \( \omega = 1 - \nu(0) \frac{\mu}{\lambda} \).

### 4.3.3 Asymptotic Analysis of the Retrieval Time of an Estimate

In this section, we determine \( \mathbb{E}[W_s] \) for asymptotically large rate of energy harvesting, battery capacity and size of the sensor network. First, we introduce the following lemma.

**Lemma 4.4.** For any \( 0 \leq k \leq s, k \in \mathbb{N}, \)

\[
\binom{N}{k} \sum_{v=0}^k \binom{k}{v} (-1)^{k-v} \frac{N-k}{N-v} = 1. \tag{4.11}
\]

**Proof.** This proof follows from induction on \( k \).
4.3 Analysis

It is easy to see that (4.11) holds for \( k = 0 \). We assume that (4.11) holds for some \( k > 0 \). We next show that (4.11) holds for \( k + 1 \).

\[
\left( \frac{N}{k+1} \right) \sum_{v=0}^{k+1} \binom{k+1}{v} (-1)^{k+1-v} \frac{N - (k+1)}{N-v} \]

\[
= \left( \frac{N}{k+1} \right) \sum_{v=0}^{k+1} \binom{k+1}{v} (-1)^{k+1-v} \frac{N - v + v - (k+1)}{N-v} \]

\[
= \left( \frac{N}{k+1} \right) \sum_{v=0}^{k+1} \binom{k+1}{v} (-1)^{k+1-v} (1)^v \]

\[
+ \left( \frac{N}{k+1} \right) \sum_{v=0}^{k+1} \binom{k+1}{v} (-1)^{k+1-v} v - (k+1) \frac{N - v}{N-v} \]

\[
= 0 + \left( \frac{N}{k+1} \right) \sum_{v=0}^{k+1} \binom{k+1}{v} (-1)^{k+1-v} \frac{(k+1) - v}{N-v} \]

\[
= \left( \frac{N}{k+1} \right) \sum_{v=0}^{k} \binom{k+1}{v} (-1)^{k-v} \frac{k!}{N-v} \frac{1}{(k+1-v-1)!} (-1)^{k-v} \frac{1}{N-v} \]

\[
= \left( \frac{N}{k} \right) \sum_{v=0}^{k} \binom{k}{v} (-1)^{k-v} \frac{N-k}{N-v} = 1, \]

where the last equality follows from the induction hypothesis.

\[\square\]

**Theorem 4.3.** For \( 1 \leq N < \infty \) and \( 0 < B < \infty \),

\[
\lim_{\lambda_e \to \infty} \mathbb{E}[W_s] = \sum_{j=0}^{s-1} \frac{1}{\mu(1-j/N)}. \]

**Proof.** Taking \( \lambda_e \to \infty \) in Theorem 4.2, we have that

\[
\lim_{\lambda_e \to \infty} \mathbb{E}[W_s] = \sum_{j=0}^{s-1} \left( \frac{N}{j} \right) \sum_{v=0}^{j} \binom{j}{v} (-1)^{j-v} \frac{1}{\mu(1-v/N)}. \]

The result now follows from Lemma 4.4. \[\square\]
Similar results to that of Theorem 4.3 have been obtained in Chapter 2, Part II for a model where a network of $N$ wireless sensors broadcast their measurements following the decentralized $D$ schedule according to which, at an exponential rate $\mu$, a random sensor is scheduled for transmission. In contrast to the current model, in Chapter 2, Part II, we assumed that sensors always have energy to broadcast. In the current model we assume an energy harvesting model for the sensors and, thus, sensors may not always have energy to broadcast. However, assuming $\lambda_e \to \infty$, the batteries of the sensors are, with high probability, always full and, therefore, the model considered in this chapter reduces to the model addressed in Chapter 2, Part II for the decentralized $D$ schedule.

We next consider the situations when the capacity of the sensors to store energy in the battery is asymptotically large.

For $\lambda_e < \mu/N$ and $B \to \infty$, the battery of a sensor is most of the time empty as the rate at which this sensor receives energy is lower than the rate at which this sensor broadcasts, and thus, consumes energy. As a consequence, in this case, the waiting time for a client to retrieve $s$ measurements from distinct sensors largely depends on $\lambda_e$, which supports the broadcasting process.

For $\lambda_e > \mu/N$ and $B \to \infty$, a sensor has most of the time energy for broadcasting since the rate at which it harvests new energy is higher than the rate at which it broadcasts. In this case, the waiting time for a client to retrieve $s$ measurements from distinct sensors depends on the broadcasting rate $\mu/N$.

**Theorem 4.4.** For $1 \leq N < \infty$ and $0 < \lambda_e < \infty$,

$$
\lim_{B \to \infty} \mathbb{E}[W_s] = \begin{cases}
\sum_{j=0}^{s-1} \frac{1}{\lambda_e(N-j)}, & \lambda_e < \frac{\mu}{N} \\
\sum_{j=0}^{s-1} \frac{1}{\mu(N-j)}, & \lambda_e \geq \frac{\mu}{N}.
\end{cases}
$$

**Proof.** We first consider the case $\lambda_e < \mu/N$. Then, from Lemma 4.1, $\lim_{B \to \infty} \nu(0) = 1 - \frac{\lambda_e}{\mu/N}$. Using this result in Theorem 4.2 we have that

$$
\lim_{B \to \infty} \mathbb{E}[W_s] = \sum_{j=0}^{s-1} \binom{N}{j} \sum_{k=0}^{j} \binom{j}{k} (-1)^{j-k} \frac{1}{\lambda_e(N-k)}.
$$

The result follows from Lemma 4.4.
4.4 Numerical Results

We next consider the case $\lambda_e \geq \mu/N$. Then $\nu(0) \to 0$. Using this result in Theorem 4.2 we have that

$$\lim_{B \to \infty} \mathbb{E}[W_s] = \sum_{j=0}^{s-1} \frac{N}{j} \sum_{k=0}^{j} \binom{j}{k} (-1)^{j-k} \frac{1}{\mu(N-k)/N}.$$ 

Again, the result follows from Lemma 4.4.

**Theorem 4.5.** For $0 < B < \infty$ and $0 < \lambda_e < \infty$,

$$\lim_{N \to \infty} \mathbb{E}[W_s] = \frac{s}{\mu}.$$ 

**Proof.** Recall that a measurement is broadcasted at an exponential rate $\mu$ from the sensor network, for any $N > 0$. Moreover, given that $N \to \infty$, the probability that there is always at least one sensor with energy which can transmit, tends to 1. Also, the probability that any $s$ consecutive broadcasts are from distinct sensors, tends to 1 as $N \to \infty$. From the above arguments it follows that, for each of the $s$ measurements, a client waits, in expectation, $1/\mu$. The result follows.

4.4 Numerical Results

In this section we analyze numerically the expected waiting time for a client to retrieve a reliable estimate of an attribute under various assumptions concerning the size of the wireless sensor network, the maximum battery capacity of the sensors and the rate at which sensors harvest energy from the environment.

Figures 4.3a and 4.3b show $\mathbb{E}[W_s]$ under various $N$, the size of the wireless sensor network. As expected, as $N$ increases, $\mathbb{E}[W_s]$ decreases. The reason is that, as $N$ increases, the probability that at least one sensor has battery to broadcast an innovative measurements increases. Thus, it is expected that clients wait less to retrieve $s$ measurements to compute a reliable estimate of an attribute. Moreover, Figures 4.3a and 4.3b show that, for a fixed $\lambda_e$, if $B$ is increased, then $\mathbb{E}[W_s]$ decreases. This is because as $B$ increases, more energy can be collected in the sensor batteries, which enables broadcasts.

Figure 4.3a considers the case when $\lambda_e \geq \mu/N$, whereas Figure 4.3b considers the case when $\lambda_e < \mu/N$. When $\lambda_e \geq \mu/N$, it is expected that most of the time the batteries of the sensors have energy. If $\lambda_e < \mu/N$, the batteries
4. DATA RETRIEVAL TIME FOR ENERGY HARVESTING SENSORS

(a) $\lambda_e = 0.2$, $\mu = 0.4$, $s = 2$.

(b) $\lambda_e = 0.03$, $\mu = 0.4$, $s = 2$.

Figure 4.3: $\mathbb{E}[W_s]$ under various $N$, the size of the sensor network.

Figure 4.4: $\mathbb{E}[W_s]$ under various $B$, the maximum battery capacity of a sensor, $N = 10$, $\mu = 0.4$, $s = 2$.

are expected to be empty most of the times. This explains the fact that $\mathbb{E}[W_s]$ takes lower values in Figure 4.3a than in Figure 4.3b.

Figure 4.4 shows $\mathbb{E}[W_s]$ under various $B$, the battery capacity of a sensor, and for various $\lambda_e$, the rate at which sensors harvest energy from the environment. As expected, for a fixed $B$, $\mathbb{E}[W_s]$ decreases as $\lambda_e$ increases. This is the case because the battery of the sensors are more frequently replenished and, thus, the sensors have energy to broadcasts their measurements. We note that for $\lambda_e \in \{0.1, 0.2\}$, $\lambda_e > \mu/N$, while for $\lambda_e \in \{0.03, 0.02\}$,
\( \lambda_e < \mu/N \). Figure 4.4 also shows that, for a fixed \( \lambda_e \), if \( B \) increases, then \( \mathbb{E}[W_s] \) decreases. This decrease becomes less significant for large values of \( B \). This can be explained as follows. In the case that \( \lambda_e \geq \mu/N \), even though sensors are able to store large amounts of energy, i.e. \( B \) is large, the rate at which the sensors broadcast is low and thus, \( \mathbb{E}[W_s] \) mostly depends on the broadcasting rate, rather than \( B \). In the case that \( \lambda_e < \mu/N \), even though \( B \) is large, the amount of energy in the batteries is expected to be low most of the times. Thus, in this case, the fact that \( B \) is very large does not result in a significantly decrease in \( \mathbb{E}[W_s] \).

### 4.5 Conclusions

In this chapter, we considered the problem of retrieving a reliable estimate of an attribute from a wireless sensor network that harvests energy from the environment. We assumed that energy is available for harvesting at each sensor according to a Poisson process. Moreover, the sensors store the harvested energy in a limited-capacity battery. Provided there is sufficient energy stored, the sensors broadcast their measurements, independently, at random.

We showed that the time until an arbitrary sensor broadcasts has a phase-type distribution. Based on this, we determined the probability distribution of the time to retrieve from the sensor network a reliable estimate of an attribute. We also provided a closed-form expression for the expected time to retrieve this estimate. In addition, we determined the retrieval time of a reliable estimate when the energy available for harvesting, the storage capacity of the sensor battery or the size of the sensor networks is asymptotically large.

Lastly, we analyzed numerically the retrieval time of a reliable estimate under various assumptions concerning the size of the wireless sensor network, the maximum capacity of the batteries of sensors, as well as the rate at which sensors harvest energy from the environment. The numerical results show that deploying sensors with very large batteries does not result in a significant decrease in the retrieval time of a reliable estimate. On the other hand, the numerical results show that the rate at which sensors harvest energy and the rate at which they broadcast significantly influences the retrieval time of a reliable estimate.
4. DATA RETRIEVAL TIME FOR ENERGY HARVESTING SENSORS
CHAPTER 5

Caching: Deployment vs. Data Retrieval Costs

5.1 Introduction

Similar to the settings in Chapters 2, 3 and 4, in this chapter we consider the problem of retrieving data from a fixed number of locations. However, in this chapter we assume that a data file needs to be retrieved from wireless caches which store this file in a distributed fashion. In particular, we analyze the relation between the storage capacity of the caches, the deployment strategies used for the caches, and the retrieval process of data from the caches.

The model considered in this chapter is as follows. Consider wireless caches placed in the plane according to a homogeneous spatial Poisson process. A client, arriving at a random location in the plane, is interested in retrieving a large data file that is stored at the caches. We assume that the storage capacity of the caches is limited. Thus, the data file needs to be stored in a distributed fashion, i.e. each cache stores a fragment of the data file. Consequently, the client needs to retrieve data fragments from several caches to recover the complete file.

Data fragments can be stored at the caches according to various strategies. We study two storage strategies: partitioning and coding [67]. Partitioning is a storage strategy according to which the data file is divided into equally-sized fragments. Replicas of these data fragments are stored at the caches. In the coding strategy, each cache stores a random linear combination of the fragments.

We focus on two cost measures and their Pareto front. The first cost measure is the deployment cost of the caches in the plane, defined to be proportional to the storage capacity of the caches and the density of the spatial Poisson process according to which the caches are deployed. The second cost measure is the cost for a client to retrieve the complete data file from the caches. This is the cumulative cost of obtaining individual data fragments
from specific caches. The retrieval cost is increasing in the distance between the client and the contacted caches. Figure 5.1 illustrates how the retrieval cost is affected by the density of the caches.

Figure 5.1: The influence of data storage strategy and cache density on the data retrieval and cache deployment costs. Partitioning: the file consists of fragments $A, B, C$. Coding: fragments are independent linear combinations of $x_1, x_2, x_3$. A client needs to retrieve the complete data file from the caches. High cache density enables shorter distances between the client and the transmitting caches. This example shows that in the case of low cache density, the distances between the client and the transmitting caches, based on which the retrieval cost is computed, are expected to be higher (double arrow) than in the case of low cache density. Therefore, the retrieval cost decreases at the expense of higher deployment costs.

One could reduce the retrieval cost by increasing the density of the caches, i.e., the average number of caches per unit area. In this case, however, the deployment cost would increase. A trade-off arises between the deployment of the caches in the plane and the cost of retrieving the complete data file. In this chapter, we analyze the Pareto front of the deployment and data
retrieval costs under partitioning and coding storage strategies. The parameters over which we optimize are:

i) the density of the spatial Poisson process according to which caches are deployed in the plane;

ii) the storage capacity of an individual cache.

Data replication and coding for caches have been studied in [30, 31, 38–40, 51, 65, 75, 107]. In [51], the authors consider the optimal number of replicas of data such that the distance between a node requesting data and the nearest replica is minimized. In [39, 40], the problem of which files to store at which caches based on the files’ popularity is analyzed. A client requests content from a neighborhood of caches. The aim is to maximize the probability that a client’s request is satisfied given the available files in its neighborhood of caches. Data sharing among multiple caches such that the bandwidth consumption and the data retrieval delay are minimal, is considered in [107]. In [38], the optimal collaboration distance between caches is analyzed. In [31] coded data allocation at the caches is investigated such that any sufficiently large subset of caches can provide the complete data. In [65] network coding for caches is employed, where users have access to a single cache and a direct link to the source. It is shown how coding reduces the load on the link between the caches and the source. The impact of non-orthogonal transmissions for coding is considered in [75]. Scaling results on the best achievable transmission rates are derived. In this chapter, we assume that data transmissions from the caches to the client are orthogonal, by separating them in time, frequency or coding space.

Cache deployment in the plane under the partitioning and coding storage strategies has been studied in [7]. It is shown that, for fixed cache density and cache storage capacity, coding dominates partitioning, i.e. the retrieval cost under coding is always smaller than under partitioning. Our interest is in both optimizing the cache deployment density and the storage capacity of the caches. This has motivated us to investigate the Pareto front of the cache deployment and data retrieval costs.

In this chapter, we provide an exact characterization of the Pareto front of the cache deployment and data retrieval costs under the coding and partitioning storage strategies. In particular, we show that for any Pareto optimal partitioning strategy there exists an optimal coding strategy that dominates it. We also show that under partitioning, the Pareto front depends only on the ratio of the deployment density and the storage capacity of the caches. Thus, when deploying caches, one has some flexibility in either increasing the density or the capacity of the caches. For coding, however, we show
that it is optimal to deploy many caches with low storage capacities. Thus, even though the optimal coded strategy leads to lower costs, it provides less flexibility in deployment.

Moreover, we quantify the improvements offered by the optimal coding in comparison to the partitioning strategy, i.e., we derive a relation for the difference in deployment cost required to achieve a given retrieval cost. Finally, we show that even non-optimal coding is better than the partitioning strategy in the sense that no coded deployment can be dominated by a partitioning strategy.

The remainder of this chapter is organized as follows. In Section 5.2 we define the model and formulate an exact problem statement. We compute the Pareto front for the retrieval time and the cache deployment cost in Section 5.3. In Section 5.3.3 we provide a comparative analysis of the performance of the coding and partitioning strategies. In Section 5.4 we provide final remarks and conclusions.

5.2 Problem Statement

We consider a data file of \( n \) symbols, with \( n > 1 \) fixed, which is stored at the caches. The symbols are elements of a finite field \( \mathbb{F}_q \).

Wireless caches are placed in the plane according to a homogeneous spatial Poisson process with density \( \lambda_d \), where \( \lambda_d > 0 \) is a parameter over which we optimize. The caches have limited storage capacity which we express in terms of an integer \( k \), the second optimization parameter. A cache stores fragments of a file of size \( n/k \) symbols. To ensure integral cache capacity, we impose that \( n \) is divisible by \( k \), denoted by \( k | n \). Therefore, \( 1 \leq k \leq n \) and \( n/k \) is an integer.

A client arriving at a random location in the plane is interested in retrieving the complete data file from the caches. We assume that the client has complete knowledge about the content and the location of the caches. The client requests data from a set of \( k \) caches that ensures the recovery of the complete data file.

Under the partitioning strategy (Part), the file is divided into \( k \) different fragments, each of \( n/k \) symbols. Each cache selects uniformly at random a fragment to store. A client requests fragments from \( k \) closest caches such that all \( k \) fragments are distinct. If a specific fragment is not allocated at any of the caches, then we reallocate the file fragments at the caches.

Under coding (Code), each cache stores a random linear combination of the \( k \) fragments. The closest set of \( k \) caches is chosen to decode the file.
There is a positive probability that \( k \) random linear combinations are not linearly independent and, therefore, do not provide the entire data file. In this case, the client needs to request data fragments from caches that are located further away. It was demonstrated in [7] that this has a negligible impact on the retrieval cost. Therefore, we restrict our attention to the cost of retrieving the data from the \( k \) nearest caches.

The cost measures, which are a function of the model parameters \( k \) and \( \lambda_d \), are defined as follows:

i) The expected data retrieval cost, denoted by \( C_r^B \), with \( B \in \{ \text{Part, Code} \} \).

Let the cost of retrieving data from \( k \) caches located at distances \( \delta_1, \ldots, \delta_k \) be

\[
C_r^B(\delta_1, \ldots, \delta_k) = \frac{1}{n} \sum_{i=1}^{k} \frac{n}{k} \delta_i^{2\alpha} = \frac{1}{k} \sum_{i=1}^{k} \delta_i^{2\alpha},
\]

(5.1)

where \( \alpha \geq 1/2 \) is an arbitrary, but fixed, parameter.

This cost function can denote, for instance, the power needed for a cache to transmit data to a client. Indeed, from the capacity of a AWGN channel \( 1/2 \log(1 + P\delta^{-2\alpha}) \), where \( 2\alpha \) is the path loss exponent in the wireless medium, the power to transmit at a guaranteed minimum rate \( G \) is

\[
P = (e^{2G} - 1)\delta^{2\alpha}.
\]

In (5.1), \( \frac{n}{k} \delta_i^{2\alpha} \) is the cost of retrieving \( n/k \) symbols from a cache at distance \( \delta_i \) away from the client. We normalize the retrieval cost by \( n \), as seen in (5.1). We are interested in the expected cost \( C_r^B(k, \lambda_d) \), where the expectation is over the randomness in the spatial Poisson process according to which the caches are deployed in the plane.

ii) The expected deployment cost of the caches in the plane per unit area, denoted by \( C_d \).

The cost of deploying a single cache is proportional to the cost of storing \( n/k \) symbols and the deployment density \( \lambda_d > 0 \) of the caches in the plane. We normalize the cost by \( n \). The expected deployment cost per symbol per unit area is defined as:

\[
C_d(k, \lambda_d) = \frac{\lambda_d}{k},
\]

(5.2)

where the expectation is over the randomness in the spatial Poisson process according to which the caches are deployed in the plane and the randomness of the storage strategy.

We consider the multi-objective optimization problem which aims at minimizing the expected deployment cost \( C_d(k, \lambda_d) \) and the expected retrieval cost \( C_r^B(k, \lambda_d) \) under the storage strategy \( B \in \{ \text{Part, Code} \} \).
We will use the gamma function, which for $x > 0$ is represented as
\[
\Gamma(x) = \int_0^\infty t^{x-1}e^{-t} \, dt,
\]
and the digamma function, defined as (see, for instance, [1]),
\[
\psi(x) = \frac{d}{dx} \ln \Gamma(x) = \int_0^\infty \left( \frac{e^{-t}}{t} - \frac{e^{-xt}}{1-e^{-t}} \right) \, dt.
\]

5.3 Analysis

In general, a single point simultaneously minimizing two conflicting objectives does not exist, in which case the multi-objective problem does not have a unique optimal solution. Thus, we characterize the Pareto front [22] of the expected cache deployment and data retrieval costs. The Pareto front shows to what extent one of the objectives can be improved at the expense of the other.

First, we introduce the following definitions [22]. Let $F_1$ and $F_2$ be two objective functions of a multi-objective optimization problem.

**Definition 5.1.** Pareto Dominance
If $x$ and $y$ are both feasible, we say that $x$ dominates $y$ if $F_i(x) \leq F_i(y)$, $i \in \{1, 2\}$, and for at least one $i$, $F_i(x) < F_i(y)$.

**Definition 5.2.** Pareto Optimality
A Pareto optimal point $x^{PO}$ satisfies the following: if $y$ is feasible and $F_i(y) \leq F_i(x^{PO})$ for $i \in \{1, 2\}$, then $F_i(x^{PO}) = F_i(y)$, with $i \in \{1, 2\}$. In words, a point is Pareto optimal if and only if it is feasible and there is no better feasible point.

**Definition 5.3.** Pareto Front
The set of Pareto optimal values for a multi-objective function is called the Pareto front.

Multi-objective methods such as scalarization or goal programming are commonly used to compute the Pareto front. We refer to [62] for an extensive survey on methods to compute optimal Pareto solutions. In this chapter, the specific structure of the objective functions allows us to directly determine their Pareto front.
5.3 Analysis

5.3.1 Partitioning

In the following theorem we determine the expected cost of retrieving the complete data file from the caches, when the partitioning strategy is used.

**Theorem 5.1** ([7]). The expected cost of retrieving the data file under the partitioning strategy is:

\[ C_{\text{Part}}^r(k, \lambda_d) = \left( \frac{k}{\lambda_d \pi} \right)^\alpha \Gamma(\alpha + 1). \] (5.3)

Next, we determine the set of points that describe the Pareto front of the cache deployment and data retrieval costs under partitioning strategy.

**Theorem 5.2.** The Pareto front of the retrieval cost \( C_{\text{Part}}^r(k, \lambda_d) \) and deployment cost \( C_{\text{Part}}^d(k, \lambda_d) \) under the partitioning strategy is described by the following set of points:

\[ \{(x, y) \mid x > 0, y = \frac{\Gamma(\alpha + 1)}{(\pi x)^\alpha}\}. \]

**Proof.** The proof follows from writing the expected retrieval cost (5.3) as a function of the expected deployment cost (5.2). \( \square \)

The above results demonstrate that the Pareto front of the costs \( C_{\text{Part}}^r(k, \lambda_d) \) and \( C_{\text{Part}}^d(k, \lambda_d) \) only depends on the ratio of the optimization parameters \( x = \lambda_d/k \).

5.3.2 Coding

In the following theorem we determine the expected cost of retrieving the complete data file from the caches, when the coding strategy is used.

**Theorem 5.3** ([7]). The expected cost of retrieving the data file under the coding strategy is:

\[ C_{\text{Code}}^r(k, \lambda_d) = \frac{1}{k} \left( \frac{1}{\lambda_d \pi} \right)^\alpha \frac{\Gamma(\alpha + 1 + k)}{(\alpha + 1)\Gamma(k)}. \] (5.4)

We first state the following lemma.

**Lemma 5.1.** \( v\psi(v + s) - v\psi(v) - s < 0, \) for \( 1 \leq v \leq n \) and \( s > 1. \)
Proof. Let \( f(v, s) = v\psi(v + s) - v\psi(v) - s, s > 1 \) and \( 1 \leq v \leq n \). We show that \( f(v, s) < 0 \).

\[
f(v, s) = v\psi(v + s) - v\psi(v) - s \\
= v \int_{0}^{\infty} \left( \frac{e^{-t}}{t} - \frac{e^{-(s+v)t}}{1-e^{-t}} \right) dt - v \int_{0}^{\infty} \left( \frac{e^{-t}}{t} - \frac{e^{-vt}}{1-e^{-t}} \right) dt - s \\
= v \int_{0}^{\infty} \frac{e^{-vt}(1-e^{-st})}{1-e^{-t}} dt - s \\
= \int_{0}^{\infty} ve^{-vt} \frac{(1-e^{-st})}{(1-e^{-t})s} s dt - s \\
= \int_{0}^{\infty} h(s,t)(ve^{-vt}) s dt - s,
\]

where we defined \( h(s,t) = \frac{(1-e^{-st})}{s(1-e^{-t})} \).

First, we show that \( h(s,t) < 1 \) for \( s > 1, t > 0 \). This follows from

\[
\frac{\partial h(s,t)}{\partial s} = \frac{e^{-st}(1+ts) - 1}{s^2(1-e^{-t})} < \frac{e^{-st} \cdot e^{st} - 1}{s^2(1-e^{-t})} = 0,
\]

where, for the last inequality we used \( 1 + ts < e^{st} \). Thus, \( h(s,t) \) is a decreasing function in \( s \) and \( \lim_{s \to 1, t \to 0} h(s,t) = 1 \). Consequently, \( h(s,t) < 1 \).

Now, since \( h(s,t) < 1 \) and \( \int_{0}^{\infty} ve^{-vt} = 1 \),

\[
f(v, s) = \int_{0}^{\infty} h(s,t)(ve^{-vt}) s dt - s < 0.
\]

We are now able to determine the set of points that describe the Pareto front of the cache deployment and data retrieval costs under coding strategy.

**Theorem 5.4.** The Pareto front of the retrieval cost \( C_{\text{Code}}^r(k, \lambda_d) \) and deployment cost \( C_{\text{Code}}^d(k, \lambda_d) \) under the coding strategy is described by the following set of points:

\[
\left\{ (x,y) \mid x > 0, \ y = \frac{\Gamma(\alpha + 1 + n)}{(\alpha + 1)\Gamma(n)(\pi x)^{\alpha n^{\alpha+1}}} \right\}.
\]

Proof. Let \( x = C_{\text{d}}(k, \lambda_d) \), with \( x > 0 \), and \( y = C_{\text{Code}}^r(k, \lambda_d) \). Using Theorem 5.3, the expected retrieval cost as a function of the expected deployment cost is as follows:

\[
y = \frac{\Gamma(\alpha + 1 + k)}{(\alpha + 1)(\pi x)^{\alpha} \Gamma(k)k^{(1+\alpha)}}. \quad (5.5)
\]
5.3 Analysis

Let
\[ g(k, \alpha) = \frac{\Gamma(\alpha + 1 + k)}{(\alpha + 1)\pi^\alpha \Gamma(k)k^{(1+\alpha)}}. \]

Then \( y = g(k, \alpha)x^{-\alpha} \). We now show that \( g(k, \alpha) \) is minimized when \( k = n \). Computing the gradient of \( g(k, \alpha) \) with respect to \( k \) gives
\[
\frac{\Gamma(\alpha + 1 + k)}{(\alpha + 1)\pi^\alpha \Gamma(k)k^{(2+\alpha)}} \left[ k\psi(k + 1 + \alpha) - k\psi(k) - 1 - \alpha \right],
\]
which by Lemma 5.1 is negative. It follows that \( g(k, \alpha) \) is decreasing in \( k \). Therefore, \( y \) is minimized for \( k = n \).

Taking \( k = n \) in (5.5) gives the desired result.

The above results show that under coding, it is Pareto optimal to always take \( k = n \), i.e. to fragment the data as much as possible. Note that this result satisfies the integrability constraint of the storage capacity of the caches.

Figure 5.2 shows the Pareto front of the deployment and retrieval costs under partitioning and coding. The figure shows that optimizing for one of the objectives necessarily influences the other. For example, a low data retrieval cost for the client can be achieved at the expense of a more dense, yet more expensive, cache deployment.

![Figure 5.2: The Pareto front for coding and partitioning, \( \alpha = 2 \).](image)
5.3.3 Performance Comparison - Coding

We have shown in [69] that any point on the partitioning Pareto front can be achieved by parameters \( k = n \) and a corresponding cache density \( \lambda_d^* \). Also, it was shown that for each such point on the Pareto front there exists a point on the optimal coding Pareto front achieved by parameters \( k = n \) and \( \lambda_d^\ast \) which is dominating. Thus, any point on the partitioning Pareto front is dominated by a point on the optimal coding Pareto front.

In this subsection we quantify the performance gap between the optimal coding and the partitioning strategies. In particular, we consider for a given expected retrieval cost \( y \) the difference \( \Delta_d(y) \) of the optimal expected deployment cost under optimal coding and the optimal expected deployment cost under partitioning. More precisely, let the pairs \((k_{\text{Part}}, \lambda_{d_{\text{Part}}})\) and \((n, \lambda_{d_{\text{Code}}})\) be Pareto optimal under partitioning and coding, respectively. Moreover, let \( C_{d_{\text{Part}}}(k_{\text{Part}}, \lambda_{d_{\text{Part}}}) = C_{d_{\text{Code}}}(n, \lambda_{d_{\text{Code}}}) = y \). Then

\[
\Delta_d(y) = C_{d_{\text{Part}}}(k_{\text{Part}}, \lambda_{d_{\text{Part}}}) - C_{d_{\text{Code}}}(n, \lambda_{d_{\text{Code}}}).
\]

Similarly, we introduce the corresponding performance gap for the deployment costs

\[
\Delta_r(x) = C_{r_{\text{Part}}}(k_{\text{Part}}, \lambda_{d_{\text{Part}}}) - C_{r_{\text{Code}}}(n, \lambda_{d_{\text{Code}}}),
\]

for the case where \((k_{\text{Part}}, \lambda_{d_{\text{Part}}})\) and \((n, \lambda_{d_{\text{Code}}})\) are optimal and satisfy

\[
C_{d_{\text{Part}}}(k_{\text{Part}}, \lambda_{d_{\text{Part}}}) = C_{d_{\text{Code}}}(n, \lambda_{d_{\text{Code}}}) = x.
\]

**Theorem 5.5.** The performance gap between optimal partitioning and coding is

\[
\Delta_r(x) = b_r x^{-\alpha}, \quad \Delta_d(y) = b_d y^{-1/\alpha},
\]

where

\[
b_r = \pi^{-\alpha} \Gamma(\alpha + 1) \left( 1 - \tilde{f}(n, \alpha) \right) \tag{5.8}
\]

\[
b_d = \pi^{-1} \Gamma(\alpha + 1)^{1/\alpha} \left( 1 - \tilde{f}(n, \alpha)^{1/\alpha} \right) \tag{5.9}
\]

\[
\tilde{f}(n, \alpha) = \frac{\Gamma(\alpha + 1 + n)}{n^\alpha \Gamma(\alpha + 2) \Gamma(n + 1)}. \tag{5.10}
\]

Moreover,

\[
0 \leq b_r \leq \pi^{-\alpha} \Gamma(\alpha + 1) \left( 1 - \Gamma(\alpha + 2)^{-1} \right), \tag{5.11}
\]

\[
0 \leq b_d \leq \pi^{-1} \Gamma(\alpha + 1)^{1/\alpha} \left( 1 - \Gamma(\alpha + 2)^{-1/\alpha} \right). \tag{5.12}
\]
Proof. The results follow from (5.3), (5.4), the fact that \( \tilde{f}(n, \alpha) \) is decreasing in \( n \), \( \lim_{n \to \infty} \tilde{f}(n, \alpha) = \Gamma(\alpha + 2)^{-1} \) and \( \lim_{n \to 1} \tilde{f}(n, \alpha) = 1 \).

For a large-path loss exponent, which is the case in, for instance, millimeter-wave based communications, the performance gap between the coding and the partitioning strategies with respect to the retrieval cost can be arbitrarily large since \( \lim_{\alpha \to \infty} \Delta_r(y) = \lim_{\alpha \to \infty} \Gamma(\alpha + 1)^{1/\alpha} y^{-1/\alpha} \pi^{-1} n^{-1} = \infty \). In the case of the deployment cost, the performance gap is:

\[
\lim_{\alpha \to \infty} \Delta_d(y) = \lambda_{\text{Code}}^d (1 - \frac{1}{n}), \text{ where } y = C_{\text{Code}}^r(n, \lambda_{\text{Code}}^d).
\]

The results follow from (5.3), (5.4) and \( \lim_{\alpha \to \infty} \tilde{f}(n, \alpha)^{1/\alpha} = 1/n \).

### 5.3.4 Non-optimal Coding vs. Partitioning

In this section we show that there exists no domain of the optimization parameters where partitioning dominates coding in terms of retrieval and deployment costs, even with a non-optimal parameter choice for coding. More precisely, consider the coding strategy and allow the deployment to be non-optimal, i.e., we consider deployment of caches with parameters \( \lambda_{\text{Code}}^d \) and \( k_{\text{Code}} \), where we allow \( k_{\text{Code}} < n \). We show that there exists no deployment scenario, i.e. no positive values \( \lambda_{\text{Part}}^d, k_{\text{Part}} \), under which partitioning dominates coding. Formally,

**Theorem 5.6.** There exists no positive values \( \lambda_{\text{Part}}^d, \lambda_{\text{Code}}^d \) and \( 1 < k_{\text{Part}}, k_{\text{Code}} \leq n \) such that

\[
C_{\text{Part}}^d (k_{\text{Part}}, \lambda_{\text{Part}}^d) \leq C_{\text{Code}}^d (k_{\text{Code}}, \lambda_{\text{Code}}^d)
\]

and

\[
C_{\text{Part}}^r (k_{\text{Part}}, \lambda_{\text{Part}}^d) \leq C_{\text{Code}}^r (k_{\text{Code}}, \lambda_{\text{Code}}^d),
\]

with at least one of these inequalities holding strictly.

**Proof.** From \( C_{\text{Part}}^d (k_{\text{Part}}, \lambda_{\text{Part}}^d) \leq C_{\text{Code}}^d (k_{\text{Code}}, \lambda_{\text{Code}}^d) \) and \( C_{\text{Part}}^r (k_{\text{Part}}, \lambda_{\text{Part}}^d) \leq C_{\text{Code}}^r (k_{\text{Code}}, \lambda_{\text{Code}}^d) \) it follows by the definition of the deployment cost that

\[
1 \leq \frac{k_{\text{Part}} \lambda_{\text{Code}}^d}{k_{\text{Code}} \lambda_{\text{Part}}^d}
\]

and from (5.3) and (5.4) that

\[
\frac{k_{\text{Part}} \lambda_{\text{Code}}^d}{k_{\text{Code}} \lambda_{\text{Part}}^d} \leq \left( \frac{\Gamma(\alpha + 1 + k_{\text{Code}})}{\Gamma(k_{\text{Code}} + 1)(k_{\text{Code}} \alpha \Gamma(\alpha + 2))} \right)^{1/\alpha},
\]
respectively.

Strict inequality in (5.13) and (5.14) holds only if $C_{d, \text{Part}}(k_{\text{Part}}, \lambda_{d, \text{Part}}) < C_{d, \text{Code}}(k_{\text{Code}}, \lambda_{d, \text{Code}})$ and $C_{r, \text{Part}}(k_{\text{Part}}, \lambda_{d, \text{Part}}) < C_{r, \text{Code}}(k_{\text{Code}}, \lambda_{d, \text{Code}})$, respectively. The result follows from the observation that

$$\frac{\Gamma(\alpha + 1 + k)}{k^\alpha \Gamma(\alpha + 2) \Gamma(k + 1)} < 1, \text{ for any } k > 1.$$

Observe that if $k = 1$, then both under the partitioning and coding strategy, the entire data file is stored at each cache. In this case, there is no difference between the partitioning and coding strategy and clearly, the data file is not stored at the caches in a distributed fashion. Thus, in the analysis above we consider $k > 1$.

### 5.4 Conclusions

In this chapter, we considered the problem of retrieving data from wireless caches which are placed in the plane according to a homogeneous spatial Poisson process. Due to their limited storage capacity, these caches store a data file in a distributed fashion. We analyzed the relation between the storage capacity of the caches, the deployment strategy used for these caches and the retrieval of data from the caches, under coding and partitioning data storage strategies.

We determined the Pareto front of the expected deployment cost of the caches in the plane and the expected cost for a client to retrieve a data file stored at the caches according to coding or partitioning storage strategies. For the partitioning strategy, we derived a simple relation for the Pareto points. For the coding strategy, we showed that it is Pareto optimal to maximize the data fragmentation. We showed that storing data according to the coding strategy results in a lower Pareto front than in the case of partitioning.

We also quantified the additional cost incurred by partitioning in comparison to optimal coding when a specific retrieval or deployment cost is given. Lastly, we showed that even non-optimal coding is better than the partitioning strategy, i.e., no coded deployment can be dominated by a partitioning strategy.
CHAPTER 6

Query Processing in Wireless Sensor Networks

6.1 Introduction

In the previous chapters we analyzed various aspects of data retrieval from a wireless sensor network. In this chapter we consider the setting where data can be retrieved by querying either the wireless sensor network (WSN) or a central database (DB), where sensed data is stored periodically. In particular, we are interested in developing mechanisms to respond to queries in a timely manner and with relevant, fresh data.

We consider the situation where end-users/applications initiate queries about an attribute of the environment that wireless sensors monitor. This attribute could be, for instance, the temperature in a room. We analyze the cost of query processing seen as a trade-off between two Quality of Service (QoS) requirements:

i) the response time of queries, which is defined as the time between the moment this query is initiated (by an end-user/application) and the moment this query is solved;

ii) the age of the data provided to the queries, which is defined as the time between the moment sensed data is generated by sensors and the time this data is provided to a query.

Query response time is a significant QoS measure for sensor networks, especially in the case of real-time applications. Keeping track of the age of the data ensures that queries are provided with relevant data, which can closely characterize the monitored environment.

In recent years, studies on sensor networks have focused on energy efficient data transmission [26, 59, 72]. Growing interest in applications which have specific QoS requirements has created additional challenges. We refer to [26, 55] for an extensive outline of WSN specific QoS requirements. Significant work has been done in the past years in the area of QoS-based routing
protocols within the sensor network. Most such protocols satisfy end-to-end packet delay [44] or data reliability requirements [27, 90] or a trade-off between the two [34] in sensor networks. In [105], a query optimizer is used to satisfy query delay requirements. In [53], the authors use data validity restrictions to specify how much time is allowed to pass since the last sensor acquisition so that the sensors are not always active and some previously sensed data is used.

We analyze in this chapter the cost of query processing seen as a trade-off between the response time of queries and the age of the data provided to these queries. The system we consider consists of a central database (DB) and a WSN, both able to solve queries (see Figure 6.1).

Queries are assigned for processing either to the WSN or to the DB. When a query is assigned to the WSN, the query service of the network requests a sensor value from the network. Upon receiving this request, the network sends its most recent value.

We assume a processor sharing service type for the WSN, which reflects the IEE 802.15.4 MAC design principle of distributing the processing capacity fairly among the jobs present in the network (see also [53]). When a query is assigned to the DB, this query is provided with data which was stored at a previous time in the DB. Data is stored in a DB upon reporting. Data reports are periodically generated by the WSN. After reports are processed by the WSN, they update the DB with the reported values.
Querying the WSN is energy-expensive, slow and can overload the network, leading to high query response times. The advantage, however, is that the sensors provide queries with the most recently sensed data. Querying the DB is fast and the energy consumption is low. However, the stored data is possibly outdated, as the age of the stored data increases over time. In this chapter we provide an approach on how to trade-off between the response time of queries and the age of the stored data, in an optimal and computationally tractable manner. In particular, we determine an optimal query assignment strategy that minimizes the processing cost of the queries by trading-off between the response time of the queries and the age of the data provided to the queries.

We formulate the query assignment problem presented above as a Continuous Time Markov Decision Process (CTMDP) with a drift. The continuous character of the process, and in particular, the fact that the age component of the process evolves continuously over time, makes the problem non-standard and computationally intractable, i.e., the standard way of deriving an optimal policy recursively using dynamic programming is not directly applicable. Therefore, for computational reasons, we argue a Discrete Time and Discrete State Markov Decision Process. First, we propose a non-standard exponentially uniformized Markov Decision Process, which we show to be stochastically equivalent to the original Continuous Time Markov Decision Process with a drift. However, the exponentially uniformized process still contains the age as a continuous state component. Therefore, for further computational tractability, we argue a Discrete Time and Discrete State Markov Decision Process. We then determine an optimal query assignment policy for the discrete time and discrete state process by means of standard stochastic dynamic programming. Finally, we show that the optimal policy also holds for the original Continuous Time Markov Decision Process with a drift.

In addition, we provide a numerical comparative analysis of the performance of the optimal policy and of several heuristics. The heuristics are simple assignment strategies, derived from practice. The results provide a formal support for the design and implementation of query assignment policies in practice so that the system can perform close to the optimum.

This chapter is structured as follows. In Section 6.2, we describe the model of the query assignment problem. In Sections 6.3, 6.4, 6.5 we define this query assignment problem as a Discrete Time and Space Markov Decision Problem. In Section 6.7, we assess numerically the performance of our proposed assignment policy. In Section 6.8 we compare the performance of
our proposed assignment policy with other feasible heuristics. In Section 6.9 we discuss the comparative analysis of the proposed assignment policy and the heuristics. Concluding remarks are stated in Section 6.10.

6.2 Model Description

We consider a system (see also Figure 6.2) consisting of a wireless sensor network (WSN) with Processor Sharing capabilities and a central storage facility (DB).

We also assume two types of jobs: queries and reports. Queries arrive at the system at an exponential rate $\lambda_1$. Reports arrive at the system at an exponential rate $\lambda_2$. Reports are requests issued to the WSN only to sense the environment and send the sensed data to the DB. Reports update, therefore, the DB.

The service requirements of the jobs are exponentially distributed with parameter $\zeta$, independently of the job type. To ensure that the system is stable, we assume that $\lambda_2 < \zeta$.

![Diagram](image)

Figure 6.2: Model incorporating a controller, the database (DB) and the wireless sensor network (WSN). The DB solves queries assigned by the controller. The WSN processes reports and queries assigned by the controller. The data stored in the DB is considered outdated if the age of the data exceeds a freshness threshold $T$. After $T$ is exceeded, the age of the data increases linearly until a report completion updates again the DB and the age of the stored data is reset to zero.

Incoming queries are handled by a controller which assigns them either to the DB or to the WSN.

When queries are assigned to the DB, they are immediately answered with stored data. However, the stored data might be outdated, i.e., the age of the data might exceed a predefined freshness threshold $T$. After threshold $T$ is exceeded, the quality of the data, i.e., how fresh the data is, deteriorates...
linearly in time. When queries are assigned to the WSN, they wait to receive the sensed data, sharing the service with the other jobs present in the sensor network. We assume a Processor Sharing service type of service for the WSN. Therefore, the expected response time of the queries is growing linearly with the number of jobs being processed by the WSN (a direct consequence of Little’s Law).

The system introduced above assumes a query processing cost which is influenced by the type of query assignment (DB or WSN assignment). The cost of a DB assignment is an instantaneous cost, indicating how much the age of the stored data has exceeded the freshness threshold $T$. The cost of a WSN assignment consists of penalties, accumulating over time, related to having queries waiting in the WSN to be processed. These penalties increase upon a WSN assignment, as a consequence of the Processor Sharing service type of the WSN. When a new query arrives at the controller, the model must decide between increasing the query processing cost of the system with an instantaneous DB-related cost or with a WSN-related cost.

We are interested in finding an optimal query assignment strategy, which specifies when it is optimal to assign incoming queries to the DB or the WSN based on a trade-off between the response time of the queries and the age of the data provided to the queries. As such, in the next section, we formulate our problem as a Markov Decision Problem.

### 6.3 Continuous - Time Markov Decision Process with a Drift

In this section we formulate the system in Section 6.2 as a Continuous Time Markov Decision Process with a drift. For an introduction to Continuous Time Markov Decision Processes with a drift, see, for instance, [48].

Firstly, at any point in time, the system is completely described by the number of queries, reports and the age of the data stored in the DB. Thus, the state space of the problem is defined as follows.

- State space $S = \mathbb{N}_0 \times \mathbb{N}_0 \times [0, \infty)$, where $(i, j, t) \in S$ denotes the state with $i$ queries and $j$ reports present in the WSN, and the time $t$ since the last report completion (age of the stored data).

Upon a query arrival, the controller assigns the query for processing either to the DB or to the WSN. The action space is, thus, defined as follows.
• Action: the controller takes an action \( d \) from the action space \( \mathcal{A} = \{D,W\} \), where \( d = D \) denotes a DB assignment and \( d = W \) denotes a WSN assignment.

We define a policy \( \pi \) to be a mapping from the state space \( S \rightarrow \mathcal{A} \), which specifies the action \( d \in \mathcal{A} \) the controller takes when the system is in state \((i,j,t) \in S \) and a query arrival occurs. We make the natural assumption that this policy is right-continuous in the age component \( t \), which allows for threshold-type of assignment policies of the form \( t \geq T \), where \( T \) is a threshold.

The system has a state transition upon a query arrival, a report arrival, a query completion or a report completion. The rates at which these events happen are as follows.

• The transition rates, when the system is in state \((i,j,t) \in S \) and action \( d \in \mathcal{A} \) is taken, are as follows:

\[
q^d[(i,j,t),(i,j,t)'] = \begin{cases} 
\lambda_1, & (i,j,t)' = (i + 1,j,t), d = W \\
\lambda_1, & (i,j,t)' = (i,j,t), d = D \\
\lambda_2, & (i,j,t)' = (i,j+1,t) \\
\zeta \phi_1(i,j), & (i,j,t)' = (i-1,j,t), i > 0 \\
\zeta \phi_2(i,j), & (i,j,t)' = (i,j-1,0), j > 0 
\end{cases}
\]  

(6.1)

with \( \phi_1(i,j) = \frac{i}{i+j} \), \( \phi_2(i,j) = \frac{j}{i+j} \) indicating the Processor Sharing service discipline assumed for the WSN. The first line of (6.1) models a query arrival under action \( d = W \), i.e. the query is assigned to the WSN for processing. The state space illustrates an increment in the number of queries from \( i \) to \( i + 1 \). The second line of (6.1) models a query arrival under action \( d = D \), i.e., the query is assigned to the DB. In this case, the query is processed immediately, no changes occur in the number of the queries and reports in the system. The third line of (6.1) models a report arrival. The state of the system illustrates an increment in the number of reports. The fourth line of (6.1) models a query completion at the Processor Sharing rate \( \phi_1(i,j) = \frac{i}{i+j} \). The number of queries in the system is decremented to \( i - 1 \). Lastly, the fifth line of (6.1) models a report completion at the Processor Sharing rate \( \phi_2(i,j) = \frac{j}{i+j} \). The age of the stored data is reset to zero as a report is completed and updates the DB with the most recently sensed data.

The above Markov Decision Process has a deterministic drift for the age component, \( t \). This increases linearly as long as no report is completed. Also,
we consider two types of decisions. Firstly, the decision to assign an incoming query to the DB affects only the infinitesimal generator (see, for instance, [81]) of the Continuous Time Markov Decision Process (see second line of (6.1)). Secondly, the decision to assign a query to the WSN affects both the infinitesimal generator and determines a change in the state of the system (see first line of (6.1)).

The dynamics of this controlled Markovian decision process are uniquely determined by its infinitesimal generators (see, for instance, [33], Chapter 1). In the case of our system described above, under action $d$, the generator is specified, for any function $f: \mathbb{N}_0 \times \mathbb{N}_0 \times (0, \infty) \rightarrow \mathbb{R}$, as follows:

$$H^d f(i, j, t) = \sum_{(i,j,t)'} q^d[(i, j, t), (i, j, t)'] \cdot f[(i, j, t)'] + \frac{d}{dt}f(i, j, t) \quad (6.2)$$

The generator stated in (6.2) shows that, over time, one of the following two situations can happen:

i) a jump to a new state $(i, j, t)'$ occurs at rate $q^d$ and the time increases;
ii) no jump occurs and the time increases.

We next consider the cost of processing incoming queries at the WSN or at the DB. The query processing cost of the system is two-fold.

i) The cost $i$ of having $i$ queries waiting within the WSN to be processed. This cost indicates the load of the WSN over time. Having a large number of queries in the WSN incurs penalties as the queries need to wait more to be processed.

ii) An instantaneous cost incurred every time a query is processed by the DB. We incur a penalty for each time unit the age of the stored data exceeds a given threshold $T$.

The two costs illustrate the trade-off between the response time of the queries within the WSN and the age of the data provided to the queries.

• Cost: when in state $(i, j, t)$, a cost rate $i$ for the queries waiting in the WSN and an instantaneous cost $(t - T)^+$, where $x^+ = \max(x, 0)$, upon a DB assignment.

The cost function above assumes no explicit communication times. When queries are assigned to the WSN, the communication time is implicitly included in the time the query waits to be processed. In the case of a DB assignment, the processing and communication time are negligible compared to the time a query is processed within the WSN. Therefore, we assume a query is immediately processed when assigned to the DB.
6.4 Exponentially Uniformized Markov Decision Process

The continuous character of the process described in Section 6.3, and in particular, the continuous age component of the process, which evolves over time, make the query assignment problem computationally intractable, i.e., the standard way of deriving an optimal policy recursively using standard stochastic dynamic programming is not applicable for a Continuous Time Markov Decision Process with a drift. More precisely, the method of uniformization, commonly used to make a Continuous Time Markov Decision Process computationally tractable, is not directly applicable due to the drift (the age component evolving over time) of our process. Uniformization, as introduced in [50], is a well-known technique used to transform a continuous time Markov jump process into a discrete time Markov process. When the state is also discrete, the process is referred to as a continuous time Markov chain (see, for instance, [37, 95]).

In [48] and [96], time discretization is applied to Continuous Time Markov Decision Processes with a drift component evolving over time. Time discretization is a somewhat similar method to uniformization. Time discretization, however, is an approximative method which leads to technically weak convergence. Moreover, it does not lead to exact computational results, as aimed in this chapter.

To be able to compute an optimal query assignment policy, we construct an exponentially uniformized Markov Decision Process, and show it to be stochastically equivalent to the initial Continuous Time Markov Decision Process with a drift. This implies that the two processes are the same in terms of expected assignment costs and policies. We next construct a Discrete Time and Discrete State Markov Decision Process, which is computationally tractable. Based on this process, an optimal query assignment policy is computed. We then show that the assignment policy computed also holds for the initial Continuous Time Markov Decision Process with a drift (Section 6.3).

In the following, we uniformize the Continuous Time Markov Decision Process with a drift described in Section 6.3. First, let \( \chi \) be an arbitrarily large finite number such that

\[
\chi \geq \lambda_1 + \lambda_2 + \zeta.
\]

Next, we construct a process which, at exponential times with parameter \( \chi \), will have a transition from state \((i, j, t) \in S\), as specified in Section 6.3, to
(i, j, t)′ ∈ S. Let u denote the exponential realization time of this transition. Then, given a transition realization of duration s, the transition probabilities under action d ∈ A, from one transition epoch to the next, become:

\[
P_d[(i, j, t), (i, j, t)′] = \begin{cases} 
\lambda_1 \chi^{-1}, & (i, j, t)′ = (i+1, j, t+u), \\
\lambda_1 \chi^{-1}, & (i, j, t)′ = (i, j, t+u), \\
\lambda_2 \chi^{-1}, & (i, j, t)′ = (i, j+1, t+u), \\
\zeta \chi^{-1} \phi_1(i, j), & (i, j, t)′ = (i, j-1, j+u), \\
\zeta \chi^{-1} \phi_2(i, j), & (i, j, t)′ = (i, j, t+u), \\
1 - (\lambda_1 + \lambda_2 + \zeta \mathbf{1}_{i+j>0}) \chi^{-1}, & (i, j, t)′ = (i, j, t+u), \\
0, & \text{otherwise.} 
\end{cases}
\]

**Theorem 6.1.** For any policy π, the exponentially uniformized Markov Decision Process and the original Continuous Time Markov Decision Process with a drift are stochastically equivalent.

**Proof.** Appendix 6.11.1

A consequence of Theorem 6.1 is that the expected assignment cost for the exponentially uniformized Markov Decision Process and the Continuous Time Markov Decision Process with a drift are the same. This, in turn, leads to the same optimal policy for the two processes.

Now observe that in the Continuous Time Markov Decision Process with a drift, the actions are only taken upon query arrivals, which occur at exponential times. In the case of the exponentially uniformized Markov Decision Process, the exponential times have parameter \(\chi\). Thus, the actions will still be taken at exponential times with parameter \(\chi\), upon a query arrival. Therefore, it is sufficient to keep track of the number of exponential phases \(A\) (Erlang distribution with parameter \(\chi\) and \(A\) phases). This allows us to restrict ourselves to a Discrete Time and Space Markov Decision Process in Section 6.5. A discrete time and space Markov Decision Process enables us to compute an optimal assignment policy in Section 6.7.

In the next section we restrict ourselves to a Discrete-Time and Discrete-Space Markov Decision Problem, with \(S = \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0\), where \((i, j, A) \in S\).
denotes the state in which there are $i$ queries, $j$ reports and $A$ steps since last report completion, i.e., the age of the data is given by the number of exponential phases $A$.

### 6.5 Discrete Time and Discrete Space Markov Decision Problem

Based on the exponentially uniformized model in Section 6.4, we formulate our assignment problem as a Discrete Time and Discrete Space Markov Decision Problem (DTMDP) as follows:

- **State space:** $S = \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0$, where $(i, j, A) \in S$ denotes the state with $i$ queries and $j$ reports present in the WSN, and $A$ is the age of the stored data, with $A$ the number of steps (exponentially distributed with uniformization parameter $\chi$) since the last report completion.

- **Action space:** Upon a query arrival, the controller takes an action $d$ from the action space $A = \{D, W\}$, where $d = D$ is a DB assignment and $d = W$ is a WSN assignment.

- **The transition probabilities, when the system is in state $(i, j, A) \in S$ and action $d \in A$ is taken, are as follows:**

$$P^d[(i, j, A), (i, j, A)'] = \begin{cases} 
\lambda_1', & (i, j, A)' = (i+1, j, A+1), \\
\lambda_1', & (i, j, A)' = (i, j, A+1), \\
\lambda_2', & (i, j, A)' = (i, j+1, A+1) \\
\zeta' \phi_1(i, j), & (i, j, A)' = (i-1, j, A+1), \\
\zeta' \phi_2(i, j), & (i, j, A)' = (i, j-1, 0), \\
1 - (\lambda_1' + \lambda_2' + \zeta' \mathbf{1}_{i+j>0}), & (i, j, A)' = (i, j, A+1) \\
0, & \text{otherwise}
\end{cases}$$

(6.3)

with $\phi_1(i, j) = \frac{i}{i+j}$, $\phi_2(i, j) = \frac{j}{i+j}$, $\lambda_1' = \lambda_1 \chi^{-1}$, $\lambda_2' = \lambda_2 \chi^{-1}$ and $\zeta' = \zeta \chi^{-1}$ as per uniformization (see Section 6.4). The first two lines of (6.3) model query
arrivals under action $d$. The third line of (6.3) models report arrivals. The fourth and fifth lines of (6.3) model query and report completions, respectively. The sixth line of (6.3) is a dummy transition as a result of the uniformization. The last line of (6.3) prohibits any other state transition. Note that every step, the age is incremented, except when a report is completed, i.e., the age is reset to zero.

- **Cost function:** The cost of the system is two-fold. Firstly, when $i$ queries are waiting to be solved within the WSN, the system incurs a cost per unit of time:

\[
i \quad (6.4)
\]

This can be interpreted as, each unit of time, the system pays one unit for each waiting query. At the end of a query’s service, the system had payed one unit for each unit of time the query was in the system, i.e. the query response time. Secondly, if an incoming query is assigned to the DB, an instantaneous penalty is incurred for exceeding the freshness threshold $T$ of the stored data:

\[
\max(A' - T)^+, (x)^+ = \max\{0, x\}. \quad (6.5)
\]

where $A' = A/\chi$ is the age of the data in time units, i.e. the number of uniformization steps multiplied by the expected length of a step. In this case, the system pays for the time the freshness threshold $T$ is exceeded.

Summarizing, we consider the cost of having queries waiting in the WSN (6.4) and the instantaneous cost associated with a DB assignment (6.5). Thus, when the system is in state $(i, j, N)$, the cost incurred per unit of time is:

\[
Cost(i, j, A) = i + \lambda_1(A' - T)^+ \mathbf{1}_{(d = D)}, \text{ where } (x)^+ = \max\{0, x\}. \quad (6.6)
\]

Equation (6.6) shows that the model assesses the trade-off between increasing the processing cost of the system by the instantaneous cost $(A' - T)$ or by accumulating $i$ units of penalties every time slot, until a change in the number of queries occurs.

**Remark:**

The number of exponential phases approximates the time until a report completion by $t + s = (A + 1) \cdot \chi^{-1}$. Also, the variance of an Erlang distribution with $N$ phases and parameter $\chi$, which is the case for our discretized age, is $\frac{A+1}{\chi^2}$. As $\chi \geq \lambda_1 + \lambda_2 + \zeta$ can be chosen arbitrarily large (see [47]), by the law of large numbers, for very large $\chi$, the distribution of Erlang$(A + 1, \chi)$
will concentrate around \((A + 1) \cdot \chi^{-1}\). Thus, for large uniformization parameter \(\chi\), the Discrete Time and Discrete State Markov Decision Process approximates the uniformized Markov Decision Process arbitrarily close.

### 6.5.1 Standard Stochastic Dynamic Programming

The quadruple \((S,A,P,Cost)\) introduced above completely describes the Discrete Time and Discrete State MDP. To determine an optimal assignment policy and to use standard dynamic programming, we define the following value function:

\[
V_n(i,j,A) := \text{minimal expected assignment cost over } n \text{ steps, given that the process starts in the state } (i,j,A).
\]

Then \(V_n(i,j,A)\) is computed recursively by means of the value iteration algorithm (see, for instance, [81] Section 8.5.1 ) as follows. First, we consider \(V_0(i,j,A) = 0\). Next, we iterate according to the value iteration algorithm and the following backward recursive equation:

\[
V_{n+1}(i,j,A) = \begin{cases} 
  i' + \lambda_1' \min \left\{ \frac{V_n(i+1,j,A+1)}{(A-T')^+ + V_n(i,j,A+1)} + \lambda_2' V_n(i,j+1,A+1) 
    + \zeta' \phi_1(i,j)V_n(i-1,j,A+1)1_{i>0} 
    + \zeta' \phi_2(i,j)V_n(i,j-1,0)1_{j>0} 
  \right. 

  \left. + [1 - (\lambda_1' + \lambda_2' + \zeta' 1_{i+j>0})] V_n(i,j,A+1) \right\} 
\end{cases} \tag{6.7}
\]

where \(i' = i/\chi\) and \(T' = T/\chi\), following uniformization. The first term of (6.7) is the cost of having \(i\) queries in service and a query assigned to either the WSN or the DB. The next three terms represent the cost incurred by a transition due to a report arrival, a query completion and a report completion, respectively. The final term is the dummy term due to uniformization.

Simultaneously with computing \(V_n(i,j,A)\), the algorithm computes a \(\epsilon\)-optimal stationary policy \(\pi_n\) which associates an optimizing action with the right-hand side of (6.7) for any state \((i,j,A)\). Given the \(\epsilon\)-optimal assignment policy, it is possible to compute the average assignment cost.

Denote the minimal average assignment cost by \(g^*\). Since the underlying Markov chain is ergodic, \(g^*\) is independent of the initial state. We approxi-
mate $g^*$ using the following bounds developed in [76]:

\[
L_n^* \leq g^* \leq L_n^{**}, \quad \text{where}
\]

\[
\begin{align*}
L_n^* &= \min [V_{n+1}(i, j, A) - V_n(i, j, A)], \\
L_n^{**} &= \max [V_{n+1}(i, j, A) - V_n(i, j, A)].
\end{align*}
\] (6.8)

In (6.8), $L_n^*$ is the minimum difference of the value function over two iteration steps, $n$ and $n + 1$, whereas $L_n^{**}$ is the maximum difference of the value function over steps $n$ and $n + 1$. For $n \to \infty$, $L_n^*$ and $L_n^{**}$ become arbitrarily close.

The optimal cost $g^*$ is computed with an accuracy $\epsilon$ by iterating the right-hand side of (6.7) for $n$ times until $L_n^{**} - L_n^* \leq \epsilon/\chi$ with $\chi$ the uniformization parameter. Then, the average assignment cost is approximated as

\[
g^* = \frac{L_n^{**} + L_n^*}{2}. \quad (6.9)
\]

It can be shown that the lower and upper bound converge in a finite number of steps (Theorem 8.5.4 [81]) to the $\epsilon$-optimal cost.

**Remark:** The approach of exponential uniformization proposed in Section 6.4 for a Continuous Time Markov Decision Process with a drift can be used to determine an optimal policy for any given cost function and for any fixed policy. In this chapter, we chose a cost function that reflects the trade-off between the response time of queries in the WSN and the age of the data provided to the queries.

### 6.6 Query Assignment Heuristics

In practice, simple assignment heuristics are employed to manage the query traffic. We consider the following three assignment heuristics, derived from practical assignment strategies:

- A fixed heuristic policy $\pi^{DB}$ that always assigns incoming queries to the DB. Upon a query arrival, the cost incurred is $(A - T)^+.$

- A fixed heuristic policy $\pi^{WSN}$ that always assigns incoming queries to the WSN.

- A heuristic policy $\pi^T$ that always assigns incoming queries to the DB if the age does not exceed the freshness threshold, i.e., $A \leq T$, and to the WSN otherwise.
In the next section, we will numerically compare the performance of the above heuristics with the performance of the optimal policy, under different parameter assumptions. The following theorem provides a closed-form expression for the expected assignment costs incurred by the $\pi^{DB}$ and $\pi^{WSN}$ heuristics when a freshness threshold $T$ is assumed.

**Theorem 6.2.** Assuming the DTMDP parameters $\lambda'_1, \lambda'_2$ and $\zeta'$, the average assignment cost of the heuristics $\pi^{DB}$ and $\pi^{WSN}$ are as follows,

\[
\text{Cost}_{\pi^{DB}} = \frac{\lambda'_1 (1 - \lambda'_2)^T}{\lambda'_2},
\]

\[
\text{Cost}_{\pi^{WSN}} = \frac{\lambda'_1}{\zeta' - (\lambda'_1 + \lambda'_2)}.
\]

**Proof.** Appendix 6.11.2

---

### 6.7 Numerical Results for the Optimal Query Assignment Policy

In this section, we compute numerically an optimal query assignment policy using the Discrete Time and Discrete State Markov Decision Process defined in Section 6.5.

Figure 6.3 shows which action is optimal when the system is in state $(i, j, A)$ and freshness threshold $T$ is assumed. We fix $A$, the age of the data, and $T$, and for every grid point $(i, j)$, i.e. $i$ queries and $j$ reports waiting in the WSN, blue indicates that it is optimal to assign an incoming query to the WSN and white to the DB.

We first analyze the optimal policy for a fixed age of the stored data, $A = 30$ (see Figures 3(a) and 3(b)). When $T$ is small (Figure 3(a)), the DB data is outdated most of the time. As a consequence, it is often optimal to send incoming queries to the WSN. If $T$ is increased (Figure 3(b)), then the DB data is considered fresh for a longer time. As a result, a DB assignment becomes more frequently optimal.

Next, we analyze the effect of increasing $A$ on the optimal policy, while $T$ is fixed. Figures 3(a) and 3(c) show the optimal policy when $T = 1$. In the case of large $A$ (Figure 3(c)), assigning a new query to the DB leads to large penalties generated by a more outdated DB data. Therefore, a query is more frequently assigned to the WSN. The same behavior is illustrated for $T = 4$.
6.7 Numerical Results for the Optimal Query Assignment Policy

(a) $T = 1, A = 30$.

(b) $T = 4, A = 30$.

(c) $T = 1, A = 50$.

(d) $T = 4, A = 50$.

Figure 6.3: Optimal Query Assignment Policy: WSN assignment (blue) and DB assignment (white). System parameters: $\lambda_1 = 0.8, \lambda_2 = 0.5$ and $\zeta = 1.8$. $A$ is the age of the data and $T$ the freshness threshold.
in Figures 3(b) and 3(d), where it is shown that a higher proportion of WSN assignments is associated with high data age.

Lastly, the optimal policy has a switching structure, i.e. once the number of jobs in the WSN reaches a certain threshold, the optimal policy starts sending incoming queries to the DB and continues to do so as the load (number of jobs) of the WSN increases. Moreover, the policy seems to be truncated at the boundary, for \( j = 0 \). This boundary effect is caused by the interaction between the number of reports \( j \) and the cost \( (A - T)^+ \), as indicated in equation (6.7), first and fourth line. Away from the boundary, the switching structure of the optimal policy is maintained.

Note that in (6.7), the two terms \( (A - T)^+ \) and \( \zeta' \phi_2(i, j)V_n(i, j - 1, 0)1_{j>0} \) are influenced by the age component \( A \). In addition, observe that the term \( \zeta' \phi_2(i, j)V_n(i, j - 1, 0)1_{j>0} \) becomes zero as \( j = 0 \). Therefore, at the boundary, the connection between \( j \) and \( A \) is affected, leading to changes in the structure of the optimal policy. Away from the boundary, the switching structure of the optimal policy is maintained. This indicates that the boundary effect is relatively small.

6.8 Simulation Results

In this section we compare the performance of the proposed assignment policy, i.e., the associated average cost \( (g^*) \), as defined in equation (6.9), with the average assignment cost of the heuristics proposed in Section 6.6 by means of a discrete event simulation.

The performance of the assignment policies is investigated under different parameter assumptions. While for \( \pi^{DB} \) and \( \pi^{WSN} \) exact results are presented in Theorem 6.2, we simulate the system to compute the average assignment costs for heuristic \( \pi^T \).

Simulations are also used to determine the fraction of time when queries are sent to the DB. Clearly, in the rest of the time, the queries are assigned for processing to the WSN. Therefore, this results give an indication of the load of the WSN over time.

6.8.1 The Influence of the Freshness Threshold

In this section we investigate the influence of the freshness threshold \( (T) \) on the query assignment cost.

Figure 6.4 shows that the optimal policy outperforms the heuristics when various freshness thresholds \( T \) are assumed. Figure 6.4a shows that, com-
pared with the heuristics, the optimal policy achieves a lower assignment cost. The cost difference is significant for small \( T \). This is of particular interest for real-time applications which are expected to require low freshness thresholds.

As expected, at the limit, \( T \to \infty \), both \( \pi^T \) and \( \pi^{DB} \) approach the optimal policy. This is explained by the fact that, for large freshness thresholds, the stored data is considered fresh for a long time. Consequently, it is frequently optimal to send an incoming query to the DB (see Figure 6.4b). Therefore, as \( T \) increases, the optimal policy and the heuristic \( \pi^T \) start indicating a DB assignment as the best assignment decision for incoming queries, i.e., both the optimal policy and heuristic \( \pi^T \) start behaving as heuristic \( \pi^{DB} \).

![Average Assignment Cost](image1)

![DB utilization](image2)

Figure 6.4: Average Assignment Cost and DB Utilization for different freshness thresholds \( T, \lambda_1 = 0.6, \lambda_2 = 0.4, \zeta = 1.2 \).

### 6.8.2 The Influence of the Job Arrival Rates

In this section we investigate the influence of the rate at which queries and reports arrive at the system on the query assignment cost.

Figure 6.5 shows that the optimal policy outperforms the heuristics when various query arrival rates (\( \lambda_1 \)) and freshness thresholds are assumed. For stability, \( \lambda_2 < \zeta \) is assumed (for policy \( \pi^{WSN} \), \( \lambda_1 + \lambda_2 < \zeta \)). Figure 6.5a shows that the average assignment cost of \( \pi^{WSN} \), which is based only on the response time of the queries, increases rapidly, as \( \lambda_1 \) increases. A high
\( \lambda_1 \) leads to a large number of jobs in the system and, correspondingly, to a high query response time. The assignment costs under policy \( \pi_{WSN} \) are independent of \( T \) (see Figures 6.5a and 6.5b).

The policies \( \pi_{DB} \) and \( \pi_T \) perform closely to the optimal policy, especially for low \( \lambda_1 \). For large \( T \), see Figure 6.5b, the stored data is considered fresh for a longer time, which leads to lower penalties related to the age of the data and, correspondingly, to lower assignment costs in comparison to the costs in Figure 6.5a.

As \( \lambda_1 \) increases, Figures 6.5a and 6.5b show a switching point when policy \( \pi_T \) results in higher assignment costs than policy \( \pi_{DB} \). This shows that when the number of incoming queries is expected to be high, it is preferred, from an assignment cost point of view, to send the queries most of the time to the DB. This enables the WSN to process reports faster so that the DB is updated often and the queries are provided with up-to-date data.

Figure 6.6 also shows that the optimal policy outperforms the heuristics when various report arrival rates and freshness thresholds are assumed. For stability, \( \lambda_2 < \zeta \) is assumed (for policy \( \pi_{WSN} \), \( \lambda_1 + \lambda_2 < \zeta \)). As expected, the assignment costs under policy \( \pi_{WSN} \) increase as \( \lambda_2 \) increases since more
jobs, on average need to be processed by the WSN. The assignment cost under $\pi_{WSN}$ is independent of $T$, see Figures 6.6a and 6.6b. Policies $\pi_{DB}$ and $\pi_{T}$ converge to the optimal policy as $\lambda_2$ increases.

For large $T$ (see Figures 6.6a in comparison to Figure 6.6b), the penalties related to a DB assignment decrease and, consequently, the assignment costs decrease. As $\lambda_2$ increases, Figures 6.6a and 6.6b show a switching point when policy $\pi_{DB}$ results in lower assignment costs than policy $\pi_{T}$. This shows that as the number of expected incoming reports increases, it is preferred, from an assignment cost point of view, to process only reports in the WSN, and, thus, to have all queries sent to the DB. Consequently, the DB is frequently updated and provides the queries with up-to-date data.

Figures 6.5 and 6.6 quantify the performance of such heuristics, given different arrival rates of the jobs and freshness thresholds. Moreover, looking at Figures 6.5 and 6.6, it is possible to optimize parameters such as $\lambda_2$ or $T$ so that the heuristics perform close to the optimum. For example, Figure 6.6 gives an indication on how large $\lambda_2$ should be, when $\lambda_1, \zeta, T$ are fixed, such that heuristic $\pi_{DB}$ performs close to the optimum.
6.8.3 The Influence of the Service Rate

In this section we investigate the influence of the service rate, $\zeta$, of the wireless sensor network, i.e., how fast are queries processed by the network, on the query assignment cost.

Figure 6.7 shows again that in the case of increasing $\zeta$, the optimal policy outperforms the heuristics in terms of average assignment costs. For stability, $\lambda_2 < \zeta$ (for policy $\pi^{WSN}$, $\lambda_1 + \lambda_2 < \zeta$).

As expected, Figure 6.7b shows the assignment cost of heuristic $\pi^{WSN}$ decreasing as $\zeta$ increases (in fact, the system behaves as an $M/M/1$-Processor Sharing queue with arrival rate $\lambda_1 + \lambda_2$, for which it is known that the expected response time of a job is $1/(\zeta - \lambda_1 - \lambda_2)$).

For large $\zeta$, all three heuristics converge to the optimal policy. Heuristic $\pi^{WSN}$ converges slowly to the optimal policy. The service rate needs to be very high so that assigning all queries to the WSN becomes optimal. Figures 6.7b and 6.7a show a switching point when heuristic $\pi^T$ starts to converge faster to the optimal policy than $\pi^{DB}$. This indicates that, for large $\zeta$, it is not optimal to assign queries only to the $DB$, but rather use the policy $\pi^T$. 

Figure 6.7: Average Assignment Cost for different processing capabilities $\zeta$, $\lambda_1 = 0.6$, $\lambda_2 = 0.4$. 

(a) Average Assignment Cost, $T = 3$. (b) Average Assignment Cost, $T = 5$. 
6.9 Discussion

The comparative analysis of different query assignment policies under various parameter assumptions in Section 6.8 provides insight into the performance of the system. For reasonably large freshness thresholds, simple heuristics such as $\pi^{DB}$ or $\pi^T$ perform well in comparison to the optimal policy. Such heuristics are particularly suitable for monitoring environments with little variation over time, e.g., temperature sensing. However, for applications with highly constrained data delivery requirements and large data variance over time, such as fire detection or $CO_2$ monitoring, $T$ is expected to be low. In this case, our proposed assignment policy shows significant cost savings in comparison to the heuristics. Moreover, simple heuristics like $\pi^{DB}$ and $\pi^T$, which require little computational resources, are preferred for implementation. The risk of implementing them is that they may perform far from the optimum. The comparative analysis of the heuristics and the optimal policy can provide support for the WSN providers to design their systems in such a way that simple heuristics, easy to implement in practice, can perform closely to the optimal policy. For example, while the arrival of queries is generally independent of the WSN service providers, the arrival rate of the reports or the freshness threshold specified for the stored data can be optimized so that simple heuristics have a close-to-optimum performance.

6.10 Conclusions

In this chapter we considered the problem of assigning queries about an attribute monitored by wireless sensors either to the wireless sensor network, or a central database for processing. In particular, we analyze the trade-off between the response time of the queries and the freshness (age) of the data provided to the queries.

We determined, using a Markov Decision Processes framework, an optimal assignment policy which assigns queries for processing either to the WSN or to a central DB. The optimal policy is based on the trade-off between the penalties related to having queries waiting to be process in the WSN and an instantaneous cost related to the age of the data stored at a central DB.

Moreover, we assessed numerically the performance of the optimal policy in comparison to several heuristics, derived from practice. We showed that the proposed model achieves significant cost savings, especially in the
case of real-time applications, where the freshness threshold is expected to be low. We compared several heuristics and the optimal policy under different parameter assumptions. The comparative analysis of the optimal assignment policy and of the heuristics provided a formal support to design query-based systems in such a way that simple heuristics perform close to the optimum. Moreover, this insight can assist the implementation of WSN applications, while ensuring a close-to-optimum performance of the system.
6.11 Appendix

6.11.1 Proof of Theorem 6.1

Proof. Uniformization is commonly used for Markov jump processes, making the problem computationally tractable. As a drift component is introduced in the present setting (the age component of our process evolves continuously over time), this is no longer standard.

In the following, we will show that the Continuous Time Markov process with a drift in Section 6.3 and the exponentially uniformized Markov process in Section 6.4 are stochastically equivalent by showing that the infinitesimal generators of the two processes are identical. To this end, we refer to the general concept of infinitesimal generator of a Markov process as introduced in [33].

Let $h(x)$ be a measurable function on some state space $E$ of a Markov process. Let $P(v, x, \Xi)$ be a transition function expressing the probability that a process which started in a state $x$ is in the set $\Xi$ at time $v$. Let $T_v h(x) = \int_E P(v, x, dy)h(y)$ denote a shift operator on the space $E$. Then the operator

$$H h(x) = \lim_{v \to 0} \frac{T_v h(x) - h(x)}{v}$$

is called the infinitesimal generator of the Markov process. The quantity $H h(x)$ can be interpreted as the mean infinitesimal rate of change of the process starting in state $x$. Moreover, the infinitesimal generator uniquely define a Markov process (see, for instance, [33], Chapter 1). Therefore, it is sufficient to show that the infinitesimal generator of the exponential uniformized Markov Decision Process and the original Continuous Time Markov Decision Process with a drift are identical.

In our setting, we consider the state $x = (i, j, t)$. Before addressing the infinitesimal generator of the exponentially uniformized Markov process defined in in Section 6.4, we first define the transition probability measure under action $d \in A$.

Let $P_{\Delta t}^d$ denote the transition probability measures over a time interval of length $\Delta t > 0$, given that at the last jump the system is in state $(i, j, t)$ and that following a upon a next jump, which occurs in the interval $\Delta t$, decision $d$ is taken and the system is in a new state.

As we implicitly made the assumption that a policy $\pi$, prescribing an action $d$ upon a query arrival, when the system is in state $(i, j, t)$, is right continuous and since the set of decisions is finite and discrete, for any state
(i, j, t) and fixed policy \( \pi \) there exists a \( \Delta t > 0 \) such that:

\[
\pi(i, j, t + u) = \pi(i, j, t) = d, \text{ for all } u \leq \Delta t.
\]

Let \( f : \mathbb{N} \times \mathbb{N} \times \mathbb{R} \) be an arbitrary real valued function, differentiable in \( t \). Then by conditioning upon the exponential jump epoch with variable \( \chi \) and for arbitrary function \( f \) we obtain,

\[
P_{\Delta t}^d f(i, j, t) = e^{-\Delta t \chi} f(i, j, t + \Delta t) \\
+ \int_0^{\Delta t} \chi e^{-u \chi} \sum_{(i', j', t')} \mathbb{P}_{\Delta t}^d[(i, j, t), (i', j', t + u)] f(i', j', t + u) du \\
+ o(\Delta t)^2 \\
= f(i, j, t + \Delta t) - \Delta t \chi f(i, j, t + \Delta t) \\
+ \Delta t \chi \sum_{(i', j') \neq (i, j)} q^d[(i, j, t), (i', j', t)] f(i', j', t + \Delta t) \chi^{-1} \\
+ \Delta t \chi [1 - q^d(i, j) \chi^{-1}] f(i, j, t + \Delta t) + o(\Delta t)^2 \\
= f(i, j, t + \Delta t) \\
+ \chi \sum_{(i', j') \neq (i, j)} q^d[(i, j, t), (i', j', t)] [f(i', j', t + \Delta t) - f(i, j, t)] \\
- f(i, j, t + \Delta t)] + o(\Delta t)^2,
\]

where we have used that \( q^d[(i, j, t), (i', j', t)] = q^d[(i, j, t + u), (i', j', t + u)] \) for any \( (i', j') \neq (i, j) \) and arbitrary \( s \). The term \( o(\Delta t)^2 \) reflects the probability of at least two jumps and the second term of the Taylor expansion for \( e^{-\Delta t \chi} \).

Hence, by subtracting \( f(i, j, t) \), dividing by \( \Delta t \) and letting \( \Delta t \to 0 \), we obtain,

\[
\frac{P_{\Delta t}^d f(i, j, t) - f(i, j, t)}{\Delta t} = \left[ f(i, j, t + \Delta t) - f(i, j, t) \right] / \Delta t \\
+ \chi \left[ f(i, j, t + \Delta t) - f(i, j, t) \right] + o(\Delta t)^2 \\
+ \sum_{(i', j') \neq (i, j)} q^d[(i, j, t), (i', j', t)] [f(i', j', t) - f(i, j, t)] \\
\to \frac{d}{dt} f(i, j, t) \\
+ \sum_{(i', j') \neq (i, j)} q^d[(i, j, t), (i', j', t)] [f(i', j', t) - f(i, j, t)] \\
= \mathcal{H}^d f(i, j, t), \text{ which is the generator in (6.2)}.
\]
Since the exponentially uniformized Markov decision process (as defined in Section 6.4) and the continuous time Markov decision process with a drift (defined in Section 6.3) share the same generators (see [33]), the two processes are stochastically equivalent.

6.11.2 Proof of Theorem 6.2

Proof. We first analyze the expected assignment cost under the policy $\pi^{WSN}$. The $\pi^{WSN}$ policy is independent of the freshness threshold. The WSN behaves as a M/M/1 Processor Sharing queue. Thus, the cost of the heuristic is given by the expected number of jobs in the WSN as follows,

$$\text{Cost}_{\pi^{WSN}} = \mathbb{E}[i]$$

$$= \frac{\lambda'_1}{\lambda'_1 + \lambda'_2} \cdot \mathbb{E}[i + j]$$

$$= \frac{\lambda'_1}{\lambda'_1 + \lambda'_2} \cdot \frac{\lambda'_1 + \lambda'_2}{\zeta - (\lambda'_1 + \lambda'_2)}$$

$$= \frac{\lambda'_1}{\zeta' - (\lambda'_1 + \lambda'_2)},$$

where (6.13) follows from writing $\mathbb{E}[i]$ as a fraction of the total expected number of queries and reports present at the WSN, with $\frac{\lambda'_1}{\lambda'_1 + \lambda'_2}$ the probability of a job being a query. Equation (6.14) follows from considering the total number of jobs present at the WSN, which can be viewed as the workload of a M/M/1 Processing Sharing queue, with job arrival rate $\lambda_1 + \lambda_2$ and service rate $\zeta$. Then, it is well known that the workload of this queue is $\frac{\lambda_1 + \lambda_2}{\zeta - (\lambda_1 + \lambda_2)}$.

Next, we analyze the expected assignment cost under the policy $\pi^{DB}$. Note that under policy $\pi^{DB}$, we have a state space with only two components $(j, A)$, where $j$ is the number of reports waiting to be processed and $A$ is the age of the data. Since all queries are processed by the DB, the component $i$ indicating the number of queries at the WSN is thus, always zero. For simplicity, instead of describing the state of the system in the form $(i, j, A)$, where $i = 0$ is the number of queries at the WSN, $j$ is the number of reports at the WSN and $A$ is the age of the stored data, we will consider the state $(j, A)$.

We define the cost of the policy $\pi^{DB}$ in terms of the limiting probabilities as follows:

$$\text{Cost}_{\pi^{DB}} = \lambda'_1 \sum_{A \geq T} \pi_{Age}(A) \cdot (A - T)^{+},$$

where (6.15)
where $\pi_{\text{Age}}(A) = \sum_{j \geq 0} \pi(j, A)$ is the long-run proportion of time that the process is in state $A$. In words, $\pi_{\text{Age}}(A)$ is the expected number of times the system is in a state with age $N$ during a cycle, divided by the expected length of a cycle, where a cycle is defined as the time between two consecutive resets of the age and no reports in the WSN, i.e., arriving at a state $(0, 0)$.

To compute $\text{Cost}_{\pi_{\text{DB}}}$, we need to determine $\pi_{\text{Age}}(A) = \sum_{j \geq 0} \pi(j, A)$.

The limiting probability $\pi_{\text{Age}}(A)$ is the solution of the following balance equations for components $j$ and $A$:

\[
\begin{align*}
\pi(j, A) &= \lambda_2' \pi(j - 1, A - 1) + (1 - \lambda_2' - \zeta')\pi(j, A - 1), j \geq 1, A \geq 1 \\
\pi(0, A) &= \pi(0, A - 1)(1 - \lambda_2'), j = 0, A \geq 1 \\
\pi(j, 0) &= \sum_{A \geq 1} \pi(j + 1, A)\zeta, j \geq 0.
\end{align*}
\]

(6.16)

The first line of (6.16) shows that we arrive in a state with $j$ reports and age $A$: 1) from a state where there are $j - 1$ reports, the age is $A - 1$ and a report arrival occurs or 2) from a state where there are $j$ reports, the age is $A$ and no event occurs. The second line of (6.16) shows that we arrive in a state with zero reports and age $A$ only from a state with zero reports, age is $A - 1$ and no report arrival occurs. Lastly, the third line of (6.16) shows that we arrive in a state with age zero and $j$ reports from a state with $j + 1$ reports, age $A \geq 1$ and a report completion occurs.

The system of equations in (6.16) can be solved after determining the limiting probability $\pi(0, A), A \geq 1$, for component age, and $\pi(j, 0), j \geq 0$, for component reports.

We first determine $\pi(j, 0)$. Let $\pi_{\text{Report}}(j) = \sum_{A \geq 0} \pi(j, A)$. Then the balance equations for component $j$, number of reports, are as follows:

\[
\begin{align*}
\pi_{\text{Report}}(0) &= \zeta'\pi_{\text{Report}}(1) + (1 - \lambda_2')\pi_{\text{Report}}(0) \\
\pi_{\text{Report}}(j - 1) &= \zeta'\pi_{\text{Report}}(j) + (1 - \lambda_2' - \zeta')\pi_{\text{Report}}(j - 1) \\
&\quad + \lambda_2'\pi_{\text{Report}}(j - 2), j \geq 2 \\
\sum_{j \geq 0} \pi_{\text{Report}}(j) &= 1.
\end{align*}
\]

(6.17)

The first line of (6.17) shows that we arrive in a state with zero reports in two ways: 1) from a state with one report, a report completion occurs and we jump to a state with zero reports or 2) from a state with zero reports and no arrival of a report occurs (no report completion is possible here since
there are no reports to be completed), thus, we remain in a state with zero reports.

From (6.17), using substitution, it follows that

\[
\pi_{\text{Report}}(k) = \left(\frac{\lambda'_2}{\zeta'}\right)^j \pi_{\text{Report}}(0), \quad j > 0.
\]  

(6.18)

Using (6.18) in the last line of equation (6.17),

\[
\sum_{j \geq 0} \pi_{\text{Report}}(j) = 1
\]

\[
\sum_{j \geq 0} \pi_{\text{Report}}(0) \left(\frac{\lambda'_2}{\zeta'}\right)^j = 1
\]

\[
\pi_{\text{Report}}(0) = 1 - \frac{\lambda'_2}{\zeta'}
\]  

(6.19)

We next determine \(\pi(0, 0)\), the long-term proportion of time that the system is in state \((0, 0)\), i.e. zero reports and age zero. Notice that we can arrive at the state \((0, A)\) only if we are in state \((0, A - 1)\) and no arrival of a report occurs.

\[
\pi(0, A) = (1 - \lambda'_2)\pi(0, A - 1) A \geq 1
\]  

(6.20)

Solving (6.20), it follows that:

\[
\pi(0, A) = (1 - \lambda'_2)^A \pi(0, 0).
\]  

(6.21)

Now, the proportion of time we are in a state with zero reports is the proportion of time we are in a state where there are zero reports and the age is \(A \geq 0\). Thus, using (6.21),

\[
\pi_{\text{Report}}(0) = \sum_{A \geq 0} \pi(0, A) = \sum_{A \geq 0} (1 - \lambda'_2)^A \pi(0, 0) = \frac{1}{\lambda'_2} \pi(0, 0).
\]  

(6.22)

From (6.19) and (6.22), we obtain:

\[
\pi(0, 0) = \frac{(\zeta' - \lambda'_2)\lambda'_2}{\zeta'}
\]  

(6.23)
We next state the balance equations for the age component $A$,

\[ \begin{align*}
\pi_{\text{Age}}(A) & = \sum_{j \geq 0} \pi(j, A) \\
& = \sum_{j > 0} (1 - \lambda_2' - \zeta') \pi(j, A - 1) \\
& \quad + \sum_{j > 0} \lambda_2' \pi(j - 1, A - 1) + (1 - \lambda_2') \pi(0, A - 1) \\
& = \sum_{j > 0} (1 - \lambda_2' - \zeta') \pi(j, A - 1) + \sum_{j \geq 0} \lambda_2' \pi(j - 1, A - 1) \\
& \quad + (1 - \lambda_2' - \zeta') \pi(0, A - 1) \zeta' \pi(0, A - 1) \\
& = \sum_{j > 0} (1 - \lambda_2' - \zeta') \pi(j, A - 1) + \sum_{j \geq 0} \lambda_2' \pi(j, A - 1) \\
& \quad + \zeta' \pi(0, A - 1) + \zeta' \pi(0, A - 1) \\
\pi_{\text{Age}}(0) & = \sum_{j \geq 0} \pi(j, 0) \\
& = \zeta' \sum_{j \geq 0} \sum_{A \geq 0} \pi(j + 1, A) = \zeta' \sum_{j \geq 1} \sum_{A \geq 0} \pi(j, A) \\
& = \zeta' \sum_{j \geq 0} \sum_{A \geq 0} \pi(j, A) - \zeta' \sum_{A \geq 0} \pi(0, A); \\
\sum_{A \geq 0} \pi_{\text{Age}}(A) & = 1.
\end{align*} \]

The second line of the system of equations in (6.24) shows that we arrive at a state with age $A$ and $j > 0$ reports either 1) from a state with $j > 0$ reports, age $A - 1$, and there is no report completion and no report arrival or 2) from a state with $j - 1 \geq 0$ reports, age $A - 1$ and a report arrival occurs. If the age is $A$ and there are no reports, $j = 0$, we arrive in this state from a state $(0, A - 1)$ and no report arrival occurs (here, a report completion is not possible). In equation (6.24), the component $\pi_{\text{Age}}(0)$ shows that we arrive at a state with age 0 and $j$ reports after a report completion.

Denoting $\pi_{\text{Age}}(A) = \sum_{j \geq 0} \pi(j, A)$, equation (6.24) now becomes:
\[\begin{align*}
\pi_{Age}(A) &= (1 - \zeta')\pi_{Age}(A - 1) + \zeta'\pi(0, A - 1), \quad A \geq 1 \\
\pi_{Age}(0) &= \zeta' \sum_{A \geq 0} \pi(A) - \zeta' \sum_{A \geq 0} \pi(0, A), \quad \text{with } \pi(A) = \sum_{j \geq 0} \pi(j, A) \\
\sum_{k \geq 0} \pi_{Age}(k) &= 1.
\end{align*}\] (6.25)

Observe that \(\sum_{A \geq 0} \pi(A) = 1\) and \(\sum_{A \geq 0} \pi(0, A) = \pi_{Report}(0) = \frac{1}{\lambda_2} \pi(0, 0)\) as per (6.22). Using these results in the second line of (6.25), it follows that:

\[\begin{align*}
\pi_{Age}(A) &= (1 - \zeta')\pi_{Age}(A - 1) + \zeta'\pi(0, A - 1), \quad A \geq 1 \\
\pi_{Age}(0) &= \zeta'[1 - \frac{1}{\lambda_2} \pi(0, 0)] \\
\sum_{A \geq 0} \pi_{Age}(A) &= 1
\end{align*}\] (6.26)

We solve (6.26) by induction, using that \(\pi(0, A - 1) = (1 - \lambda_2')A - 1\pi(0, 0)\), as per (6.21), and \(\pi(0, 0) = \frac{(\zeta' - \lambda_2')\lambda_2'}{\zeta}\), as per (6.23). Solving for (6.26), we have that

\[\pi_{Age}(A) = \lambda_2'(1 - \lambda_2')A.\] (6.27)

Using (6.27), we can now compute the cost (6.15) as follows,

\[Cost_{\pi_{DB}} = \lambda_1' \sum_{A \geq 0} \pi_{Age}(A) \cdot (A - T)^{+}\]

\[= \lambda_1' \sum_{A' \geq 0} \lambda_2'(1 - \lambda_2')A' + T \cdot A'\]

\[= \lambda_1' \lambda_2'(1 - \lambda_2')^{T+1} \sum_{A' \geq 0} (1 - \lambda_2')^{A' - 1} \cdot A'\]

\[= \frac{\lambda_1'}{\lambda_2'}(1 - \lambda_2')^{T+1}\]

\[\square\]
6. QUERY PROCESSING IN WIRELESS SENSOR NETWORKS
In this thesis, we have presented several contributions to the fields of data retrieval from wireless sensor networks, distributed data caching, as well as query processing for wireless sensor networks. Below we present a high-level overview of the main contributions of this thesis.

1. We provide closed-form expressions for the retrieval time of a fixed, predefined number of sensor measurements from a wireless sensor network, under various centralized and decentralized sensor transmission schedules. The underlying models consider aspects such as transmission interference, asleep/awake sensor cycles, sensor broadcasting, energy harvesting.

2. We develop sensor transmission schedules that ensure the retrieval of a fixed, predefined number of sensor measurements within a predefined time frame and with minimum energy consumption for the transmitting sensors.

3. We provide closed-form expressions for the Pareto front of two conflicting cost functions concerning the deployment of wireless caches in the plane and the retrieval of data from these caches, under two distributed data caching strategies: partitioning and coding.

4. We develop an optimal query assignment strategy with respect to the response time of queries and the age (freshness) of the data provided to the queries by means of a Discrete-Time and State Markov Decision Process, which is derived using non-standard, exponential uniformization of a Continuous-Time Markov Decision Process with a Drift.

The remainder of this chapter is structured as follows. In Section 7.1, we discuss in detail the contributions of this thesis in the field of wireless sensor networks. In Section 7.2 we discuss possibilities for future research.
7.1 Contributions and Concluding Remarks

The contributions of this thesis are as follows.

In Chapter 2, we provide closed-form expressions for the retrieval time of a fixed, predefined number of sensor measurements from a wireless sensor network, where sensors transmit their measurements according to an optimal, centralized schedule and a decentralized schedule. In Part I of Chapter 2, we show that, despite the contrasting characteristics of the centralized and decentralized sensor transmission schedules considered in this thesis, for large sensor networks, the decentralized schedule is an $e$ approximation of the centralized schedule in terms of the expected retrieval time of a fixed number of sensor measurements. The degree of difference between the optimal, centralized and the decentralized transmission schedule indicate the degree of improvement that practical, decentralized schedules can achieve. In Part II of Chapter 2, we introduce a novel, multi-class queue that models general broadcasting schedules. We determine the steady-state distribution of the system and, consequently, provide closed-form expressions for metrics such as the retrieval time of a reliable estimate of an attribute, workload of the sensor network and busy period of the system. As special cases, we derive similar performance metrics for two specific transmission schedules: a decentralized and a centralized broadcasting schedule. Nonetheless, our multi-class queueing model is general in the sense that it can be employed to derive the performance of various sensor broadcasting schedules.

In Chapter 3, we consider the problem of retrieving a fixed, predefined number of sensor measurements from a wireless sensor network with minimal energy consumption for the sensors and within a predefined time period. Moreover, the sensors are assumed to be deployed in the plane according to a random spatial process. We first provide a closed-form expression for the energy consumption of a simple, intuitively appealing, Greedy schedule. This expression is general in the sense that it does not depend on the process according to which the sensors are placed in the plane or on the function of the energy consumption assumed for the sensors. To schedule sensors under general time constraints and with minimal sensor energy consumption, we next formulate a stochastic dynamic programming (SDP) model, as well as an heuristic and an off-line, optimal sensor transmission schedule. We compare numerically the performance of the three transmission schedules in terms of the energy consumed by the transmitting sensors. The off-line transmission schedules results in the lowest energy consumption for the sensors. However, this schedule is hard to implement in practice.
7.1 Contributions and Concluding Remarks

as it requires information about the sleep/awake status of the sensors ahead of time. Both the SDP model and the heuristic provide on-line schedules that can be implemented in practice. The SDP schedule results in a lower energy consumption than the heuristic. However, its complexity may hinder its implementation in practice. The heuristic, on the other hand, performs relatively close to the SDP schedule in terms of the energy consumption and it is easier to implement in practice than the SDP schedule.

In Chapter 4, we extend the analysis of the retrieval time of a reliable estimate of an attribute in Chapter 2 to the setting where each sensor harvests energy from the environment at random, independently of the other sensors. We derive the distribution of the retrieval time of a reliable estimate when the energy storage capabilities of the sensors and the possibility to harvest energy from the environment are limited. We also provide a closed-form expression for the retrieval time of a reliable estimate for asymptotically large sensor capabilities to store energy and energy harvesting availability. Lastly, we determine numerically the retrieval time of a reliable estimate for various values of the following system parameters: size of the sensor network, harvesting rate, maximum battery capacity. The numerical results show that increasing the size of the sensor network significantly decreases the retrieval time of a reliable estimate. It is also shown numerically that deploying sensors with large batteries does not significantly reduce the retrieval time of a reliable estimate.

In Chapter 5, we provide exact results for the Pareto front of a deployment and a retrieval cost function for wireless caches, which are assumed to be deployed in the plane according to a spatial Poisson process. Data is assumed to be stored at the caches in a distributed fashion, according to partitioning and network coding strategies. We compare analytically the performance of the coding strategy in comparison to partitioning with respect to the cache deployment and data retrieval costs considered. We also show that partitioning is always dominated by coding with respect to the deployment and retrieval cost functions considered. Overall, the results show to what extent the data retrieval costs can be improved at the expense of the cache deployment costs and vice-versa. These insights provide support for the deployment of wireless caches in practice while taking into account the costs of retrieving data from these caches.

In Chapter 6, we consider query-based wireless sensor networks. We focus on the trade-off between the response time of queries and the quality (freshness) of the data provided to queries. We formulate a Discrete Time and Discrete State Markov Decision Process by means of a non-standard,
exponential uniformization of a Continuous-Time Markov Decision Process with a drift. We also derive an optimal query assignment policy. We compare numerically our optimal query assignment policy with several heuristics, derived from practice, with respect to the processing cost of queries. This comparative analysis provides insights into deriving simple query assignment policies that can be readily used in practice, while performing close to the optimal policy.

The mathematical models and the corresponding results provided in this thesis offer an overview of the performance of wireless sensor networks under various network design assumptions concerning sensor deployment, transmission protocols, energy consumption, sensor energy harvesting, data storage, query processing. The analysis of these models provides a formal, theoretical support for the design, implementation and operation of wireless sensor network applications related to retrieval of reliable data, query-based sensing and data caching. Nonetheless, the models and results provided in this thesis are of theoretical interest also in other areas such as spatial stochastic processes and information theory.

7.2 Future work

This section concerns future work related to models that consider the spatial deployment of wireless sensors, the heterogeneity of sensor networks, sensor collaboration, the variability of sensor measurements over time and spatial correlation of sensor measurements.

Spatial Deployment

The performance analysis of the wireless sensor networks considered in Chapters 2 and 4 is restricted to the retrieval time of a fixed, predefined number of sensor measurements. This analysis can be further extended by considering the deployment of sensors in the plane. In Chapter 3, we considered sensors deployed in the plane according to a spatial random process and, assuming a predefined time for the retrieval of measurements, we determined the sensor energy consumed to retrieve a reliable estimate of an attribute. It is of interest to extend the settings in Chapters 2 and 4 by considering a similar type of sensor deployment as in Chapter 3 and to determine how the retrieval time of a reliable estimate is affected by the spatial process according to which sensors are deployed. Moreover, since sensor energy consumption is often assumed to be an increasing function of the distance
between the transmitting sensors and the destination, it is of interest to compute the energy consumption associated with the retrieval of a fixed number of sensor measurements.

Heterogeneity

From a practical perspective, extending the models addressed in this thesis by considering heterogeneous wireless sensor networks, where processing capabilities, communication/sensing range and battery size vary across the sensors, is important as well. In this thesis, wireless sensors were assumed to have identical characteristics. In practice, however, it is often preferred to deploy a mix of a few expensive sensors, with high processing capabilities and large communication range, and many inexpensive sensors, with limited communication and processing capabilities (see, for instance, [68]). To allow for sensor heterogeneity, the models considered in this thesis can be extended by incorporating several classes of sensors, each class having different processing, communication and battery capabilities. One interesting line of research is to assess the trade-off between the cost of sensor deployment and sensor energy consumption or retrieval time of a reliable aggregate, for both homogeneous and heterogeneous wireless sensor networks.

Collaboration

In Chapters 2 and 3 sensors transmit their measurements in a decentralized manner, independently of each other, or according to a central schedule. Decentralized sensor transmission requires no information about the energy available at the sensors, whether sensors are awake or asleep or about sensor measurement redundancy, but the performance of the network with respect to data retrieval time and energy consumption is not optimal. Centralized coordination improves the overall performance of the sensor network, but it requires information about the status of the sensors and measurement redundancy, which is often difficult to acquire in practice. A third strategy could be to allow sensors to communicate and collaborate with each other. The models considered in Chapters 2 and 3 can be extended by allowing sensors to exchange with each other information about their sleep/awake mode, whether they have transmitted previously or intend to transmit in the near future. This information can be used by the sensors to avoid interference or transmission of redundant measurements. This, in turn, may minimize bandwidth consumption (which translates to energy savings) and may reduce the time for data retrieval, while avoiding central coordination.
of sensor transmissions. Such collaborative sensor transmissions can be further extended to define sensor-MAC (S-MAC) protocols. Several S-MAC protocols have been studied in, for instance, [28, 83, 106], where the focus is to reduce the energy waste due to transmission collisions, idle listening or overhearing. One interesting line of research is to design and analyse the performance of S-MAC protocols that allow to retrieve a reliable estimate of a monitored attribute, as considered in this thesis, while meeting constraints regarding the sensor energy consumption and the retrieval time of this estimate.

Variability of Measurements over Time

In Chapters 2, 3 and 4 we assumed that each sensor has one and the same measurement, which is transmitted to clients arriving, independently, at random, at the sensor network. The clients retrieve a set of measurements from distinct sensors to compute an aggregate. In practice, however, sensors take several measurements over time. It seems interesting to extend the models addressed in Chapters 2, 3 and 4 by considering a dynamic setting where the measurements of a sensor change over time. In particular, it seems feasible to consider the age of a sensor measurement, as introduced in Chapter 6, to specify how long ago a measurement was generated by a sensor. Additional constraints about the age of retrieved sensor measurements can be added to the models proposed in Chapters 2, 3 and 4. One such constraint could be, for instance, that only measurements with a maximum age of \( t \) time units, where \( t \) is a predefined threshold, can be retrieved to compute an aggregate. Moreover, it is of interest to determine to what extend the time to retrieve an aggregate from the sensor network is affected by these additional constraints on the age of the sensor measurements.

Spatial Correlation of Measurements

In Chapters 2, 3 and 4, sensor measurements are assumed to be independently and identically distributed across the sensors, each measurement being subject to additive normally distributed noise with zero mean and variance \( \sigma^2 \). However, sensor measurements are often spatially correlated, especially when the sensors are densely deployed in an area of interest. If the measurements are correlated, similarly to the settings considered in this thesis, it is of interest to retrieve measurements from a set of sensors, and not from the entire network, to ensure a certain level of data reliability/fidelity. The difficulty lies in selecting the set of sensors which will ensure the desired
level of data reliability (see, for instance, [4]). In this thesis, any set of sensors, of sufficiently large size, is able to provide measurements based on which an aggregate is computed. In contrast, if measurements are correlated, it becomes more useful to compute an aggregate based on measurements from a set of spatially separated sensors. The models on data retrieval considered in this thesis can be extended by assuming that measurements are spatially correlated and by adding restrictions on the distance between the sensors that provide measurements. The fact that sensors are spatially separated is expected to result in larger distances over which sensors transmit measurements. This, in turn, leads to a higher sensor energy consumption, if the sensor energy consumption is considered, as in this thesis, an increasing function of the distance over which transmissions occur.


Summary

Wireless sensor networks are currently revolutionizing the way we live, work, and interact with the surrounding environment. Due to their ease of deployment, cost-effectiveness and versatile functionality, sensors are employed in a wide range of areas such as environmental monitoring, surveillance or smart homes. While providing unprecedented opportunities for a variety of applications, current sensor networks face several challenges. For instance, the reliability of sensor measurements is often affected by the position of the sensors relative to the monitored object/phenomenon or the characteristics of the surrounding environment. It is, thus, often needed that several sensor measurements are retrieved so that reliable information about a monitored area is acquired. Moreover, since sensors can only function with sufficient energy, their functionality is affected by their limited energy resources. In this thesis, we employ the theory of stochastic processes and queuing, combinatorial theory, as well as optimization techniques such as stochastic dynamic programing to analyze the performance of wireless sensor networks, with a focus on data retrieval time, energy consumption and measurement reliability constraints.

The focus of this thesis is three-fold. Firstly, we analyze the time needed to retrieve a fixed number of sensor measurements from a wireless sensor network. Based on these measurements, an aggregate is obtained. We take into account aspects such as transmission interference, limited, stochastic energy availability induced by the fact that the sensors harvest energy from the environment, limited transmission bandwidth. Given these assumptions, we analyze the retrieval time of measurements under centralized and decentralized sensor transmission schedules. While the centralized schedules are optimal with respect to the retrieval time of measurements, the decentralized schedules require less coordination between the sensors and are more suitable for implementation in practice. Nonetheless, the degree of difference between the two types of schedules, which we derive in this thesis, indicates the degree of improvement that distributed schedules can achieve.

Secondly, we consider wireless caches, randomly deployed in the plane, that store a data file in a distributed manner. We provide an exact characterization of the Pareto front of two conflicting objectives concerning the cost of deploying the caches in the plane and the energy cost of retrieving the data file from these caches. We analyze the Pareto front under a partitioning and a network coding data caching strategy. Pareto dominance is proven for the network coding strategy, which shows that allowing for additional complexity for caching strategies, as is the case of network coding, leads to savings in terms of both energy and deployment costs.
Thirdly, we consider the case where sensed data is retrieved by querying either the wireless sensor network or a central database. We formulate an optimal query processing strategy with respect to the response time of queries and the quality (freshness) of the query data. To determine this optimal strategy, we employ a discrete-time Markov decision process, which is derived by non-standard, exponential uniformization of a continuous-time Markov decision process with a drift. We compare numerically the performance of this optimal policy with several, simple query processing heuristics, and show under which system parameters these heuristics perform close to the optimal with respect to the query response time and data quality (freshness).

Overall, the mathematical models and results derived in this thesis aim to provide a formal, theoretical support for the design of wireless sensor network applications related to the retrieval of reliable data, query-based sensing and data caching, with a goal of assisting the implementation of such applications.
Draadloze sensornetwerken hebben een revolutie ontketend in de manier waarop we leven, werken en omgaan met onze omgeving. De lage kosten van sensoren, hun multifunctionaliteit en het gemak waarmee ze geplaatst kunnen worden hebben ervoor gezorgd dat ze veelvuldig worden toegepast in controlesystemen, beveiliging en smart homes. Ondanks al deze nieuwe mogelijkheden die dit soort sensornetwerken ons bieden zijn er nog steeds veel uitdagingen. Zo hangt de nauwkeurigheid van de metingen van sensoren vaak af van de positie van de sensor tot het te meten object/gebeurtenis of van de specifieke eigenschappen van de omgeving. Het is daarom vaak nodig meerdere sensormetingen te verzamelen om daarmee betrouwbare informatie over een gebied te verkrijgen. Sensoren kunnen echter alleen functioneren met voldoende energie. Daarom wordt hun functionaliteit begrensd door beperkte beschikbaarheid van energie. In dit proefschrift passen we de theorie van stochastische processen, wachtrijen en combinatoriek, alsmede optimalisatie technieken, zoals stochastisch dynamisch programmeren, toe om sensornetwerken te analyseren. Onze focus ligt hierbij op de snelheid waarmee data wordt vergaard, het energieverbruik van de sensor en de betrouwbaarheid van de metingen.

Dit proefschrift bestaat uit drie delen. Ten eerste bekijken we de tijd die nodig is om een gegeven aantal metingen uit een sensornetwerk te verkrijgen. Onze analyse houdt rekening met interferentie tijdens het verzenden van gegevens, de beperkte bandbreedte, en hoeveelheid energie, die beschikbaar is. Dit laatste komt vaak doordat sensoren hun energie direct uit de omgeving moeten halen. Onder deze aannamen analyseren we de tijd die nodig is voor het verkrijgen van metingen uit sensornetwerken, voor gecentraliseerde en gedecentraliseerde zendprotocollen. Hoewel gecentraliseerde protocollen sneller zijn, vereisen de gedecentraliseerde protocollen minder coördinatie tussen de individuele sensoren. Hierdoor zijn de laatste meer geschikt voor praktische toepassingen. De grootte van het verschil tussen beide type zendprotocollen, die wij in dit proefschrift bepalen, een maat voor de mogelijke verbetering die gedistribueerd protocollen kunnen bereiken.

Ten tweede bekijken we draadloze caches die willekeurig in een omgeving zijn geplaatst. Deze caches slaan data gedistribueerd op. We beschouwen twee doelfuncties: een voor de kosten van het plaatsen van de caches en een voor de benodigde energie voor het ophalen van de data. Daarna geven we een exacte karakterisering van het Pareto front tussen deze twee doelfuncties. We analyseren dit Pareto front voor zowel een cache partitie strategie, als een netwerkcodering strategie. Voor de netwerkcodering strategie bewijzen we Pareto dominantie. Dit leidt tot de conclusie dat het toestaan van extra complexiteit voor de cachestrategie, zoals
het geval is voor netwerkcodering, een verbetering oplevert in zowel de plaatsingskosten als het energieverbruik.

Ten derde onderzoeken we de situatie waarin sensorgegevens of direct uit het netwerk kunnen worden opgevraagd of vanuit een centrale database. We geven een strategie voor het opvragen van gegevens die zowel de opvraagtijd als de kwaliteit van de gegevens, gemeten als het verschil tussen de huidige tijd en de tijd waarop de meting is gedaan, optimaliseert. Om deze strategie te bepalen maken we gebruik van een Markov beslissingsproces in discrete tijd. Dit proces hebben wij op een niet-standaard manier, via een exponentiële uniformisatie, afgeleid van een Markov beslissingsproces, met positieve drift, in continue tijd. We gebruiken simulaties om onze optimale strategie te vergelijken met andere, simpele, opvraag heuristieken en laten zo zien voor welke keuze van systeem parameters deze heuristische strategieën bijna optimaal presteren.

Samengevat bevat dit proefschrift wiskundige modellen en resultaten, die een theoretische basis vormen om te helpen bij het ontwerpen en implementeren van toepassingen van draadloze sensornetwerken, zoals het verkrijgen van betrouwbare data, het optimaliseren van opvraagstrategieën en data caching.
Mihaela Mitici was born in Ploiesti, Romania, on 10th of April 1988. In 2007, she obtained her Baccalaureat certificate at the I.L.Caragiale National College, Ploiesti, class of Mathematics, Computer Science, Bilingual English. In the same year, she started her Bachelor studies at the Faculty of Cybernetics, Statistics and Economic Informatics, The Academy of Economic Studies, Bucharest, obtaining her B.Sc. degree (cum Laude) in 2010. During her Bachelor program, she followed a six-months Erasmus program at the University of Amsterdam. In 2011 she obtained her M.Sc. degree in Operations Research from the University of Amsterdam. Her master thesis on ‘Flexible Air Traffic Scheduling’ was written in collaboration with NLR Amsterdam (The National Aerospace Laboratory). In 2011, Mihaela began her work as a Ph.D. candidate at the University of Twente, Stochastic Operations Research (SOR) group, within the RRR project (Realisation of Reliable and Secure Residential Sensor Platforms). In 2015, during a one-month research visit to Copenhagen, she worked at the Denmark Technical University. Her Ph.D. research culminates with this dissertation.