ABSTRACT

WANG, JIE. Modeling and Analysis of Mobile Data Dynamics in Heterogeneous Wireless Networks. (Under the direction of Wenye Wang and Xiaogang Wang.)

Owing to advances in wireless communication, networking and data analysis technologies, the generation, dissemination and acquisition of data are more frequent and accessible than ever. Consequently, data services, which *move* data from its generator(s) to its consumer(s) through individual connections, are quickly migrating to the edge of networks. Such wireless systems are composed of numerous devices that are different in many aspects, *e.g.*, communication technology, mobility pattern, and so on. Despite the benefits they brings, such as reducing service latency, emerging data services at the edge create new challenges to the network: heterogeneity of individuals adds to the complexity of system design, management, and performance evaluation, while proliferating end devices impose an ever-increasing demand on resources, that are already scarce at the edge. To exploit the full potential of data services, it is essential to understand the cause, governing rules, and impact of data's *mobility* in such heterogeneous wireless networks, for the benefit of data owners, service providers, and network operators.

Therefore, this dissertation is dedicated to study *mobile data dynamics*, that is, dynamic processes of mobile data. Specifically, we identify three dynamic processes, of *information*, *coverage*, and *spectrum*, as the cause, manifestation, and impact of mobile data, respectively. Then we examine these dynamics process and the governing rule of data movements, through a modeling and analysis approach to answer the following questions: *when* data move and stop, *where* data are, *how* data move, and *what* impact mobile data induce on network resources.

In particular, we first study conflicting information propagation with a novel *Susceptible-Infectious-Cured (SIC)* model to answer the *when* question. Our results reveal the impact of network topology on the lifetime of the undesired information, which provides bounds, scaling laws, and guidelines for practical information control measures. For the *where* question, we quantify the whereabouts of data, that is, data coverage, with a data-strength metric, and find the change of data coverage depends heavily on user mobility, based on which we establish a framework to predict data coverage, and achieve over 80% accuracy in tests with real-world traces. Then, we consider dissemination processes of multiple data blocks in the emerging DSA-enabled fog paradigm, to answer the *how* and *what* questions. We propose a *gravity model* to describe how data move in an offloading process, based on which we find that, the amount of storage and communication resource needed for data offloading scales linearly with the network size. Particularly for the spectrum resource, a scarce resource in wireless networks, we study *spectrum activity surveillance (SAS)* to observe the impact of mobile data, and propose multi-monitor deployment strategies with guaranteed performances for both the dedicated and crowd-source SAS scenarios. The work in this dissertation advances our understanding on mobile data, and provides design guidelines for data services in heterogeneous wireless networks.

Modeling and Analysis of Mobile Data Dynamics in Heterogeneous Wireless Networks

by
Jie Wang

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Engineering

Raleigh, North Carolina

2019

APPROVED BY:

Huaiyu Dai

Min Kang

Wenye Wang
Co-chair of Advisory Committee

Xiaogang Wang
Co-chair of Advisory Committee

DEDICATION

To curiosity,
which drove me this far.
To my families,
who are with me, even when we are apart.

BIOGRAPHY

Jie Wang grew up in Luoyang city, Henan, China. She received her B.S. and M.S. degree in Electrical Engineering from Tongji University, Shanghai, China, in 2010, and 2013, respectively. After her graduation with honor, Jie worked as a software designer in Ericsson Shanghai R&D center, from 2013 to 2014. In August 2014, she joined the Ph.D. program at North Carolina State University in the Department of Electrical and Computer Engineering. Her research focuses on modeling data dissemination processes in networked systems, and analysis of fundamental properties of such dynamic processes.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Dr. Wenye Wang, and my co-advisor. Dr. Xiaogang (Cliff) Wang. It is a great honor of mine to have the opportunity to work closely with Dr. Wenye Wang, and learn from her vision, passion, wisdom, and most of all, her attitude toward life. I feel so fortunate to have her as my advisor, who enlightens me when I am challenged by research problems, and patiently helps me regain confidence when I am frustrated over difficulties. She will always be my role model, as a female professional, that I wish I could live up to some day in the future. I sincerely thank Dr. Cliff Wang for his suggestions, encouragement, and guidance during my Ph.D. study. I believe that the knowledge and skills they have imparted to me are far beyond this dissertation.

I would like to extend my gratitude to my committee members, Dr. Min Kang and Dr. Huaiyu Dai, for their valuable feedback and comments, which significantly improved the quality of my research. Many thanks to all the professors, who have taught me in and out classes here at NC State, so that I could be prepared for my future career.

I gratefully acknowledge the help and support from my fellow labmates during my Ph.D. study: Zhuo Lv, Yujin Li, Mingkui Wei, Sigit Pambudi, Rui Zou, Yali Wang, Tanli Xu, Teng Fei, Cheng Chen, and LoyCurtis Smith. Studying and Working with them at the NetWIS lab has been a pleasant memory, which I will always cherish throughout my life. I would also like to recognize the assistance from the administrative personnel in the department of ECE, who have been very friendly, and helped me on numerous forms and procedures.

Finally, I would like to give my special thanks to my families, for their unconditional love and unwavering support: my parents Enping Wang and Xinde Wang, who always believe in me, encourage me to see the world, and support me both emotionally and financially; my beloved husband Bin Li, who stands by me, cares about me, and shares laughter with me throughout this process; and my grandma Meifang Liu, who taught me to be an independent, upright, and caring person by setting good examples herself. My gratitude also goes to my friends Juan Xiao, Zhiyuan Li, Hui Guan, Shaoliang Nie, and Ruiyuan Jia, for our enduring friendship.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AP access point x, 3, 54, 57, 60, 72, 74, 77, 82, 85

AR augmented reality 87

ARMA auto regressive moving average 75, 76

BS base station 3, 54, 61, 85

CDF cumulative (probability) density function 35, 97, 98, 125

CDN content delivery network 55

CR cognitive radio 109, 110

D2D device-to-device communication 3, 54

DSA dynamic spectrum access 3, 5, 9, 108–112, 116, 123, 127, 132, 147, 148, 150

ER Erdös-Rényi random graph 29, 36

ETC electric toll collecting 54

GFT graph Fourier transform 67, 69, 75

GSP graph signal processing 53, 56, 66, 73

IoT Internet-of-Things 1, 3, 5, 10–12, 15, 54, 55, 57, 68, 81, 82, 85, 93, 150

ITS intelligent transportation systems 54, 76, 81

JWSS joint wide-sense stationary 67, 73–75

LAN local area network 3, 85

LBS location-based service 53, 54

LTE Long Term Evolution 3, 54, 86

MACC mobile ad hoc cloud computing 81

MCC mobile cloud computing 81, 82

MEC mobile edge computing 81, 82

MFA mean field approximation 13, 65, 77

MGF moment generating function 32

OBU on-board unit 54

OSN online social network 1, 10–12, 24, 29, 44

PDF probability density function 96, 101, 102

PU primary user 113

QoS Quality-of-Service 83

RAT radio access technology 3, 86

RSU roadside unit 3, 54

SAS spectrum activity surveillance vii, xi, 8, 108–112, 116–119, 123, 124, 126, 127, 134, 135, 137, 139, 141–145, 147–150

SI Susceptible-Infected 13, 28, 38

SIC Susceptible-Infected-Cured ix, 14–16, 18–24, 26–28, 32, 33, 44, 45, 47, 52

SIR Susceptible-Infected-Recovered 14, 28

SIS Susceptible-Infected-Susceptible 13, 28, 38

SU secondary user 113, 114, 118

VR virtual reality 87

Chapter 1

Introduction

1.1 Motivation

Data, which refers to *transmittable and storable computer information*, has been an integral part of modern society, since the invention of computers. Especially in the past decade, its indispensable role in various applications, ranging from marketing [126] to scientific researches [40], has been re-enforced by advances in data mining, machine learning, and artificial intelligence studies. As the proliferation of smart devices that can interconnect with each other on-the-go, the creation, collection, and analysis of data are much easier and more accessible than before, leading to huge amounts of data being generated in both wired and wireless networks almost every second. For instance, the online social network (OSN) giant Facebook has 1.6 billion daily active users, who generate more than 4 PetaByte new data every day [127]. Meanwhile, the delivery networks of data are evolving into large, complex systems, due to the explosive growth of wireless devices, *e.g.*, the number of Internet-of-Things (IoT) devices is expected to exceed 500 billion by 2030 [30], imposing a tangible impact on mobile data traffic. Consequently, recent years are witnessing the transition of data from a commodity owned by companies, to a *service* that can be provided/acquired by anyone, just like the transition of computing resource in cloud infrastructure a decade ago [102]. The principal course of such service is to *move* data from its generator(s) to its consumer(s) through a network of data carriers. In this *data dissemination* process, data is *mobile*, in the sense that both the traffic volume and whereabouts are constantly changing due to user movement and data forwarding actions.

1.1.1 Data Is Alive and Mobile

From the data owner/disseminator's perspective, it is their natural rights to know who have taken (temporary) possession of its data, where those data blocks have traveled to, and when a data block of interest stops circulating in a certain region. All of these questions are tied closely to movements of data in a dissemination process.

From the delivery network's perspective, data is *alive*, that is, interacting with individuals in the network, only when it is *mobile*. In other words, the *lifetime* of data begins at the time instant when it is first injected into the network, and stops when none of its copies is circulating in the network of interest any more. During its lifetime, every move of a data block induces dynamical changes in the network, with respect to both resources and network status. In the former aspect, mobile data utilizes various kinds of network resources, for example: storage resource is consumed at a device, when the data block is temporarily stored by intermediate data carriers (for later forwarding); bandwidth/spectrum resource is consumed, when it is transmitted between data carriers; and computation resource is consumed, when it needs to be fragmented, processes, or routed. On the other hand, operating status of both individuals and the networked system as a whole, *e.g.*, capacity and system integrity, are in turn impacted by the mobile data. For example, a computer malware, *e.g.*, the SMS Trojans [43] that spreads over emails and messages, can hide in data blocks, piggyback on their dissemination process, and attack multiple users in an institutional computer network. As such a process unfolds in time, normal operations of the networked system may not be sustained.

Therefore, it is both primitive and essential to understand data's *mobility*, for the design, management, and recovery of data-delivery networks, as well as the provision of service transparency to data owners. Specifically, we identify the following open questions to understand the dynamic processes with respect to mobile data:

1. *When* does a data block start and stop moving in a network?

2. *Where* is the data block of interest accessible during dissemination?

3. *How* do data blocks move in the heterogeneous wireless network?

4. *What* is the observable impact of mobile data on data-delivery networks?

Among these, the first and second questions focus on the dissemination process of a single data block, while the rest two questions are for cases of multiple blocks. Particularly, the first question focuses on the time domain, in which the cause, or the driving force of mobile data, dictates the start and stop of the dissemination process, *i.e.*, the *lifetime* of a data block in a network. The second question focuses on the space domain, in which the *whereabouts* of data refer to the time-varying locations, where the data block (and its copies) is accessible. The third question asks for the *governing rule* of mobile data, taking interactions of multiple data dissemination processes into consideration. The last question focuses on the consequence and *impact* of data being mobile, especially on shared network resources.

Considering that movements of a single data block already create a dynamic process in the space that spans over time, geographical location, and spectrum domains, the sheer complexity of multiple data blocks being replicated, piggybacked and transmitted in the same network prohibits these questions to be answered with a single model, nor a simple solution. Therefore, we first specify and analyze the scenario of mobile data, and then introduce our solution that tackles the aforementioned problem in different domains.

1.1.2 Mobile Data Dynamics in Heterogeneous Wireless Networks

Wireless network is becoming the primary choice of many data disseminators, and the common practice of content delivery services, which are enabled by developments in wireless communication and networking technologies, including 5G [115], dynamic spectrum access (DSA) [84], IoT [4], fog computing [119], and so on. This can be observed by the ever-increasing wireless (especially mobile) traffic volume [120] and spectrum demand. For instance, ever since 2015, over 80% of social media traffic in the U.S. comes from wireless mobile devices, as well as over 50% of all website traffic world wide [62]. Such a wireless data-delivery paradigm is adopted by various application scenarios, including data sharing/forwarding [67, 68] in Long Term Evolution (LTE)-device-to-device communication (D2D) networks, mobile advertisement [100] in WiFi-LTE networks, safety message dissemination [68, 130] in vehicular networks, IoT provisioning by LTE-based fog [4], and many more. Motivated by its extensive applications, this dissertation is devoted to study data mobility in heterogeneous wireless networks, which are composed of both wireless end devices, such as mobile phones, smart vehicles, and sensory devices, and network elements of the wireless access network, such as base station (BS) in cellular networks, AP in wireless LAN, and roadside unit (RSU) in vehicular networks.

Such wireless data-delivery networks exhibit distinct characteristics, which also bring new challenges: First, data carrying individuals (users) themselves are mobile, resulting in intermittent data transmission links, and changing network topologies. Moreover, user mobility introduces the notion of ‘where’, *i.e.*, geographical location, which further complicates the problem. Second, individuals in such networks can be highly diverse in many aspects, including communication protocol, radio access technology (RAT), data forwarding preference, mobility pattern, etc., creating a dynamic and heterogeneous environment, which is difficult to model and experiment on. Last but not least, the data-delivery network can be formed in a spontaneous and ad hoc manner, which means there may not exist control of any form, unlike in a wired system with central control, *e.g.*, a cloud computing system built on Amazon EC2 servers [34].

To address these challenges, we identify three dynamic processes, each describes the behavior of mobile data in one of the three domains, namely, *time*, *geographical space*, and *spectrum*, such that their properties are tractable to be analyzed individually, and collectively they are comprehensive enough to understand mobile data in heterogeneous wireless networks. These dynamic processes reflects the cause, manifestation, and result of data’s mobility, are hence referred to as *mobile data dynamics*.

1.1.2.1 Information Dynamics: the Driving Force of Mobile Data

Any data-delivery network is designed to facilitate the flow of *information*, in the form of moving data blocks, so the beginning and the end of information propagation decide the start and stop time of data movement. However, as networks evolve into more complex systems, increasing and more diverse users introduce information from various aspects, even *conflict-*

ing information, *e.g.* rumor v.s. truth, in networks. Similar phenomena can also be observed in many networks, if the information concept is generalized to include virus/malware, system operation status and adoption of new products. Despite different manifestations, we observe that all the conflicting information propagation instances share some common characteristics: the later-injected desired information targets at an existing undesired information, stops its propagation, and terminates a potential/ongoing *epidemic* of the latter. In this way, the undesired information resembles an infectious *virus* that can infect susceptible individuals, while the desired information functions as an *antidote* that can permanently immune susceptible individuals or cure infected individuals. Due to this asymmetry, the propagation process is *transient*, in the sense that its asymptotic behavior at time t goes to infinity is known (virus-free), and the two epidemic process co-exist only for a short period of time, as opposed to the long-term coexistence (and equilibrium) in existing models on competing epidemics, *e.g.*, [94, 16, 33]. Accordingly, a natural question is, how such propagation process evolves in *finite* time. In other words, there is little knowledge on the aftermath of conflicting information propagation via individual spreading, especially *when* the undesired information (virus) dies out, *how fast* the number of victims of the virus decreases below a predetermined level, and how to design effective information (antidote) distribution strategy to reduce the lifetime of undesired information in a network. These questions have a broad impact on design, operation, and management of networks, because information propagation is the driving force of mobile data, and the extinction of the undesired information marks the end of its propagation (generation of data traffic), while the impact of the propagation reveals requirements of data delivery services.

1.1.2.2 Coverage Dynamics: the Whereabouts of Mobile Data

In a wireless data-delivery network that includes mobile devices, a data dissemination process is actually a sequence of data relocation actions, which are the superposition of data transmissions and entity (data-carrying individuals) movements. In this scenario, data's mobility is manifested in its time-varying *coverage*, that is, the geographical location where the data block of interested can be accessed (legally by its consumer or monitor, and illegally by malicious attackers). Then the direct question of mobile data follows: *where* is the data during a dissemination process? If such a dissemination process is viewed as a cause-and-effect phenomenon, existing research focus on single factors from the cause-direction, *e.g.*, from the network topology and communication aspects [100]. While most literature [100, 68, 130, 88] do not carry the notion of 'where', the ones that do [141, 66] assume entities are homogeneous in every aspect. To answer the *where* question for a heterogeneous scenario, a new model is needed to describe data's current whereabouts/coverage. Knowing the dynamically changing data coverage is important to both the data owner(s) and the delivery network. To the former, data dissemination should be transparent to the customer (data owner/disseminator) as a service, while to the later, particularly network elements, including access points, base stations and gateways, changing coverage translates to traffic load and is hence relevant to resource management, as well as policy and charging plan design.

1.1.2.3 Spectrum Dynamics: the Impact of Mobile Data in the Spectrum Domain

Spectrum is one of the most important resource in current wireless systems, due to its scarcity. Any data transmission in a wireless network will result in a spectrum *activity*, *i.e.*, occupancy of a spectrum slice for a certain period of time (typically several milliseconds) at a geographical location, so that no other individual in this region can utilize the same slice simultaneously. In other words, *spectrum dynamics*, that is, time-varying spectrum activities over a geographical region, is the *outcome* of mobile data, as well as an impact to the system capacity. To observe and evaluate such impact, it is necessary to have a spectrum surveillance system, which carries out continuous scans of spectrum activities on the frequencies of interest, for the purpose of usage data collection, including temporal and spatial patterns of spectrum occupancy, user mobility, as well as traffic patterns. Spectrum surveillance is particularly crucial to dynamic spectrum access (DSA)-enabled systems, because of the risks introduced by its open and opportunistic nature. In prior studies of spectrum surveillance strategies (*e.g.*, [107, 56, 65]), an implicit assumption is that spectrum monitors are sufficiently powerful, such that they can watch over the entire geographical region of interest and tune/move without any limit. The fact is that most spectrum activities, including communications, attacks/jamming and monitoring/sniffing, are *local*, *i.e.*, confined in both the spectrum domain and the space domain during a fixed-length time interval. This discrepancy is especially pronounced in wide-band wide-area spectrum monitoring, which naturally leads to an open question: *how to model* a spectrum surveillance process and *design* deployment strategies for monitors? Answers to this question sketch a way to evaluate the impact of data mobility in emerging wireless systems, where DSA is enabled to improve spectrum efficiency.

1.1.2.4 Network-Data Interaction: Governing Rules of Mobile Data

In addition to the three mobile data dynamics, which are external manifestations of data's movements in different domains, it is equally necessary to study *how data move*, *i.e.*, the governing rule of mobile data, to fully understand how their movements are constrained by the available resource in a wireless system, and how their movements in turn affect the system. This question is especially meaningful to emerging networking paradigms, such as fog computing [119], because they are expected to support frequent data offloading induced by IoT applications at a massive scale, with rather limited resources [78]. While existing study in this line of research design offloading schemes to direct where data blocks should move, *e.g.*, [140, 134, 135, 23, 97], to minimize the offloading latency [129] and/or energy consumption [128] for a single task, or to maximize revenue of the network [133], there is little knowledge on how to evaluate the performance of such systems, particularly how much resource the offloading process consumes, especially how it scales with the network size. In fact, scalability has been recognized as one of the major design concerns for the fog paradigm [2], and the crux of the resource consumption problem [139], which motivates us to model the data offloading processes and study the performance evaluation problem for the resource-constrained edge networks.

1.2 Research Questions and Contributions

With the three mobile data dynamics identified in different domains, and network-data interaction specified for data's mobility, we study mobile data in a *top-down* manner: we start from the driving force (information dynamics) in the time domain, then move to the description of whereabouts (coverage dynamics) in the space domain, further to the governing rules of data movements, and finally arrive at the impact on networks (spectrum dynamics) in the spectrum domain. Each step addresses one question (*when*, *where*, *how*, and *what*) identified at the beginning of this section. Next, we briefly summarize the research questions to address, in each step of this dissertation, and list our contributions toward the understanding of mobile data.

1.2.1 Information Dynamics: When Data Start and Stop Moving

Information propagation is the driving force of mobile data. With respect to information dynamics, the first work studies the propagation process of conflicting information, which is expected to answer *when data start and stop moving.* Considering the resemblance between a pair of conflicting information and a pair of virus and antidote, we model such a propagation process as two competing epidemics. To address the open question of how virus and antidote epidemics evolve in the same network, specifically, *how much time* it takes the undesired information (virus) to stop propagating, we derive bounds for such dynamics, in terms of network size and initial conditions. In this process, we reveal the influence of network topology on the lifetime of the undesired information. Further, as applications of the proposed model, we propose information injection strategies to reduce the lifetime of the undesired information, and design an algorithm to infer the time-varying number of information adopters in the network. Our contributions toward the *when* (Question 1) question are summarized as follows.

- **Modeling**: We propose a *susceptible-infected-cured* (SIC) epidemic model to study the propagation process of antidote-virus-like conflicting information in a network, which captures the competing-while-spreading effect of the virus and antidote.

- **Metric**: We define *extinction time* and *half-life time* of the virus, to describe the lifetime of the undesired information, as well as to quantify the effectiveness of information injection as a countermeasure against the undesired information.

- **Results**: We derive upper and lower bounds of the extinction time and half-life time, for SIC dynamics in networks with an arbitrary topology, which reveals how topological properties of a network change the scaling of virus' lifetime, over the size of the network.

- **Design guideline**: We propose a divide-and-conquer guideline for antidote (desired information) injection, to effectively reduce the extinction time of the undesired information.

- **Algorithm**: We design an inference algorithm, to estimate the number of information adopters in the network, during the competition between the two pieces of information, to better understand the real-time impact of the undesired information.

1.2.2 Coverage Dynamics: Where Data Are during Dissemination

The second topic focuses on the *whereabouts* of a data block (and its copies) during its dissemination process. To answer *where data are*, we formally define and quantify data coverage, that is, where the data block is accessible, such that data coverage can be represented by a time-varying signal on a *graph*, which captures physical movements of individual users/entities, and embeds geographical locations. Analyzing snapshots of the graph signal reveals the impact of user mobility, and how mobility can be used in predicting data coverage in the future. Then we build a prediction framework and evaluated its accuracy with real-world GPS traces. Our contributions toward the *whereabouts* (Question 2) of mobile data are summarized as follows.

- **Modeling**: We propose an entity model, based on which *data coverage* is formally defined, quantified by the *data-strength* metric, and formulated as a series of numeric signals on a graph, which embeds location information.

- **Metric**: We define the mobility dependence index (MDI) to quantitatively reveal the impact of user mobility on the change of data coverage. We observe high MDI when the maximum moving speed and the participating probability are low, indicating it is possible to predict data coverage based on previous observations and user mobility in these cases.

- **Application**: We build a prediction framework of data coverage, which achieves an over 80% accuracy in simulation with real-world mobility traces.

1.2.3 Governing Rules: How Data Move during Offloading

The third work focuses on the *governing rule* and the *impact* of mobile data in an offloading process, which is an exemplified application of the fog computing paradigm. With respect to *how data blocks move*, we study the inter-play of multiple tasks that compete for network resource during their offloading processes, and propose a gravity-based offloading model that can describe a variety of offloading criteria. With respect to the *impact on network resource*, we define device and network efforts, which quantify the the amount of resource consumed during the offloading of any task. Our contributions toward the *governing rule* (Question 3) and *impact* (Question 4) of mobile data are summarized as follows.

- **Modeling**: We propose a *gravity*-based offloading model, to capture probabilistic task offloading processes in the fog, by which a variety of offloading criteria can be described.

- **Metric**: We define task *lifetime*, *device effort* and *network effort*, to quantify the offloading performance for individual tasks, and the amount of storage resource, and communication resource consumed by the offloading process.

- **Result**: We derive upper bounds for the performance metrics under a generic gravity rule, and find that the lifetime of individual tasks is at most constant with the network size, and the total efforts spent by the system scale linearly with the network size.

1.2.4 Spectrum Dynamics: What Observable Impact Mobile Data Cause

Our last work focuses spectrum dynamics, which is the result of mobile data. Particularly, we study spectrum activity surveillance (SAS) processes over a large geographical region, to answer *what is the observable impact of mobile data* in the spectrum domain. To be more specific, we identify the goals of SAS to be sweep-coverage of the spectrum/space, and detection of spectrum culprits. Taking geographical space, and locality of spectrum activities into consideration, we propose a two-step solution, such that any SAS process can be formulated into a graph walk process. Two typical surveillance scenarios, namely, the dedicated and crowdsoruce scenarios, are analyzed to address their distinct design concerns, *i.e.*, efficient sweep-coverage and quick detection of culprits, respectively. As an application of the proposed modeling approach, we propose deterministic and randomized monitor deployment strategies, whose performances, *i.e.*, coverage time and detection time are analyzed. Our contributions toward the *observable impact* (Question 4) of mobile data are summarized as follows.

- **Modeling**: We model spectrum activities in a *spectra-location space* that incorporates spectra, temporal and geographical domains, while the *locality* of spectrum activities is captured. Then the SAS strategy design is formulated as a graph walk problem.

- **Metrics**: We define the *coverage time* and *detection time* for monitor deployment strategies, so that the qualitative data collection and culprit detection objectives are translated to clear quantitative metrics, by which strategies can be fairly compared.

- **Results**: We show that, despite the switching capacity limits, randomized strategies of m monitors can achieve sweep-coverage over a space of n assignment points in $\Theta(\frac{n}{m}\ln n)$ time, and detect a persistent or adversarial culprit in $\Theta(\frac{n}{m})$ time.

1.3 Organization of This Dissertation

In summary, our systematic study of mobile data dynamics expands our knowledge on data dissemination services, as they migrate to the network edge, and provide practical guidelines for fast, efficient, and predictable service provision in heterogeneous wireless networks.

This dissertation revolves around the cause, description, governing rule, and impact of mobile data, and takes a modeling approach to address the *when*, *where*, *how*, and *what* questions of mobile data, which are identified at the beginning of this chapter. The rest of this dissertation is organized as follows: Chapter 2 discusses the conflicting information propagation process, to answer when data starts and stops moving in networks. Then in Chapter 3, coverage dynamics, that is the manifestation of mobile data in a wireless environment, is introduced to examine the whereabouts of a single data block (and its copies) in its dissemination process. Chapter 4 focuses on the governing rule of mobile data, which studies how multiple data blocks move during offloading, under resource constraints in a fog computing system. Chapter 5 focuses on observing the impact of mobile data, for which spectrum activity surveillance is studied for DSA-enabled wireless systems. Finally, this dissertation is concluded in Chapter 6, where possible extensions are also discussed.

Chapter 2

Information Dynamics: Modeling and Analysis of Conflicting Information Propagation in a Finite Time Horizon

In this chapter, we study information dynamics, particularly the propagation process of conflicting information in networks, which provides in-depth understanding of how network topology determines the lifetime of mobile data dynamics. We find that the lifetime of the undesired information can be upper and lower bounded by functions of the network size and topological properties. Specifically, when connections concentrate on a few individuals to create bottlenecks in networks, the lifetime of the undesired information can change from decreasing to increasing with the network size. Taking computation complexity into consideration, we obtain upper bounds with vertex eccentricities, that is, the largest distance between an individual and any other individuals in the network, from which we propose a 'divide-and-conquer' guideline to inject desired information, such that the undesired information can be eliminated in a shorter period of time. In addition, to observe the propagation process in a finer time resolution, we provide an inference algorithm to estimate the number of information adopters, which can foresee the instantaneous evolution of the dynamics before it fully unfolds.

2.1 Introduction

Due to proliferating mobile devices and emerging mobile applications, people are more connected with each other than ever. Consequently, networked systems, such as OSN, institutional computer networks, and IoT, are developing into much larger and more complex structures than before. For instance, the OSN giant Facebook has 1.6 billion daily active users, who generate more than 4 PetaByte new data every day [127], while the number of IoT devices is expected to exceed 500 billion by 2030 [30], imposing a tangible impact on mobile data traffic. As more and more individuals, *e.g.*, users in OSN, and devices in IoT, join such systems, inconsistent, even *conflicting information* are injected into the network during the same time period, leading to an interesting competition while both pieces propagate via individual connections.

By conflicting, we mean two pieces of information that can not be admitted by the same individual at the same time. For example, it is highly unlikely, for an OSN user to simultaneously admit the truth and a rumor that contradicts with the truth. Particularly for such pairs, in which one piece (the desired information, *e.g.*, truth) is apparently more credible than the other (the undesired information, *e.g.*, a rumor), an individual who has chosen to admit the desired information, will not be affected by its undesired counterpart. Therefore, the undesired information, though spreads via individual connections itself, will be eliminated from the network given sufficient time. This phenomenon can be seen in various systems.

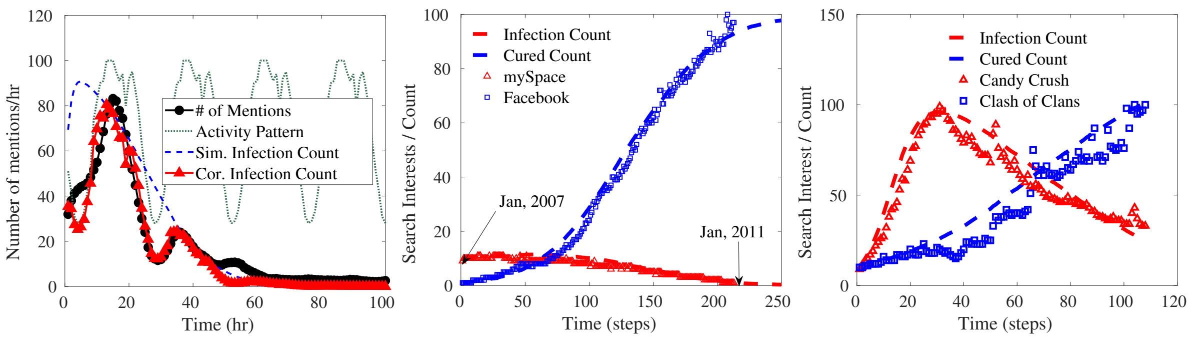
2.1.1 Motivating Examples in Different Systems

Conflicting information propagation is prevalent in social networks, *e.g.*, the word-of-mouth networks among acquaints, and OSN, which have become the arena of clashing opinions, unverified reports, and publicity campaigns. Similar competing-while-spreading phenomena can also be observed in engineered systems, such as the computer network of an institution, and IoT.

2.1.1.1 Rumor v.s. Truth in OSN

After the Boston bombing incident on April 15th, 2013, Reddit users started an online suspect hunt, which identified an innocent person as the bomber [6]. This rumor (undesired information in the form of image data) spread rapidly on both Reddit and Twitter, leading to serious cyber-harassment to the wrongly-accused. The number of mentions regarding this rumor quickly decreased after the police released correct information (desired information) [60]. Similarly in the OSN, the meme tracker App [131] recorded the number of posts with “#SpecialOlympics#” from March 20th to 24th, 2009, as shown in Figure 2.1a. At that time, President Obama made an inappropriate joke in ‘The Tonight Show with Jay Leno’, and soon apologized to the Special Olympics chairman to correct his mistake. However, it took four days for the whole incident to die down after the apology was made. In case of such epidemic spreading of both desired and undesired information, a natural question is: *when will the undesired information stop propagating*, such that it will not affect users in the network any more?

Moreover, it is observed that the lifetimes (referred to as the news cycles) of both rumors and news are becoming noticeably shorter [63, 131] than before. Is this phenomenon caused by the rapid increase of active users in the OSN [127]? What does this phenomenon imply about the topological structure of the OSN, which is too big and complex to be studied as a whole?



(a) Number of mentions in OSN: the '*Special Olympics*' Incident

(b) Popularity of OSN: MySpace v.s. Facebook.

(c) Popularity of mobile games: Candy Crash v.s. Clash of Clans.

Figure 2.1 Motivating examples of conflicting information propagation in different networks.

2.1.1.2 Advanced Product v.s. Outdated Product in Word-of-mouth Networks

When two rivalry products compete through the word-of-mouth network, where people are affected by the 'reputation' of a product among their friends, the newer and more advanced product will eventually dominate the market, by swapping out its outdated counterpart that is less appealing to customers. For instance, Facebook overturned the OSN market in its competition with MySpace, which was popular when Facebook just opens registration in January 2007, as shown in Figure 2.1b. The newer mobile game Clash of Clans, drew mobile users' attention quickly from its counterpart, Candy Crush, as shown in Figure 2.1c. In case of rivalry product competition, manufacturer of the better product will be interested in the market share (adopter of the desired information) its product takes before reaching a full market penetration, to plan and allocate business resources. In other words, how does the dynamic process evolve, particularly *how the number of adopters of different information change* over time?

2.1.1.3 Malware v.s. Security Patch in Institutional Computer Networks and Faults v.s. Restoration Commands in IoT

Engineered systems such as institutional computer networks and IoT are usually large [120], and operate under the control of system administrators. When the system administrator of an institutional computer network spots spreading malware infections, such as SMS Trojans [43] that spreads over emails and messages, and Chameleon virus [82] that spreads over WiFi links, a possible countermeasure is to distribute replicable security patches, which can fix infected computers, and also prevent uninfected computers from malware infections. Similarly in IoT systems, such as a smart grid, when cascading failures [122] are triggered by overloaded stations, the administrator can apply load shedding measures [17] to restore some affected stations, such that their neighboring stations can also benefit from the restoration, and gradually return to normal operation. In both cases, the system administrator know the structure of the networked system, and can proactively inject desired information (security patch, load shedding command), to control and eliminate the epidemic spread of undesired information (malware, fault). Hence, a practical question for the system administrator is, *where to inject the desired information* such that the undesired information (malware, failures) can die out faster? In other words, how to leverage knowledge on network topology to design more effective countermeasures against the viral spreading of undesired information?

Despite the manifestation, these conflicting information propagation processes have two defining characteristics: i) There are *two* pieces of information circulating the same network, both spread via contacts of individuals in an *epidemic* manner. ii) The later-injected desired information can convert victims of the undesired information back to normal states, just like a replicable *antidote* can cure/immune an individual from an infectious *virus*, as a result of which, the cured individual will not be infected by the same virus again, but not vice versa.

2.1.2 Related Work

Considering the great resemblance between information propagation and virus spreading, *epidemic models* have been adopted to describe the propagation process over individual connections, *e.g.*, rumors spread between friends, and malwares spread between computers, in which the information is modeled as an infectious virus. Literature on this line of research can be categorized into single-virus epidemics and multi-virus (or competing) epidemics.

Among the extensive study on single-virus epidemics [89, 122, 22, 95, 42, 61, 59], Ganesh *et.al.* [42] and Krishnasamy *et.al.* [59] identified the significant impact of the Cheeger constant $\eta(\mathcal{G})$, which measures the expansion property of the underlying network $\mathcal{G}$, on the propagation time of the virus. To be more specific, the spreading time of the virus is upper bounded by a function of $\eta(\mathcal{G})$ [59] under the Susceptible-Infected (SI) model, while for the Susceptible-Infected-Susceptible (SIS) model that allows infected individuals to recover on themselves, the virus can live for an exponentially long time with respect to network size n [42], if the recovery-to-infection rate is larger than $\eta(\mathcal{G})$. However, these single-virus models can not describe the competition between multiple pieces of information, which is the key characteristic of conflicting information propagation. In fact, the single-virus SI epidemics can be viewed a special case of the conflicting information propagation process, in the sense that it only observes the epidemic process of the antidote, which is not affected by the virus.

With respect to competing epidemics between conflicting information, existing literature can be further categorized into *population dynamics* and *network dynamics* [86], depending on whether the network topology is taking into consideration. In population dynamics, participants of the propagation process are assumed to be a well-mixed population, *i.e.*, there is no notion of network, which does not apply to most propagation scenarios. In contrast, network dynamics view participants of the propagation process as heterogeneous, and model their connections with a graph structure. Among these, Lin *et.al.* [70] utilized mean field approximation (MFA) to conduct asymptomatic and numerical analysis of the propagation process. Prakash *et.al.* [94] proposed an SI$_1$I$_2$S model, and proved that a piece information with faster propagation speed will eliminate its slower counterpart as time approaches infinity. More recently, Dadlani *et.al.*. modeled competing memes as epidemic processes on multi-layered graphs, and derived critical survival threshold of a meme [33] to be persistent. Newman [83] found the coexistence threshold of two competing epidemics on networks with known degree distributions, under the Susceptible-Infected-Recovered (SIR) model. From the perspective of propagation model, both the linear threshold model and $\mathrm{SI_1I_2S}$ model allow any individual to switch back and forth between different information pieces, in which context the asymptotic behavior (steady state of the system as time $t \to \infty$) of the dynamics is of more interest. In our case, however, the desired information is much more credible than its undesired counterpart, so competition between the two finishes in *finite time*, for which existing analysis on asymptotic behavior does not apply.

2.1.3 Our Approach and Contributions

Motivated by the lack of study, this chapter discusses the propagation process of a pair of virus-antidote like conflicting information in networked systems, in order to understand *when* the resulted mobile data dynamics starts and stops in such systems. Our contributions can be summarized as follows.

We propose a *SIC* propagation model, to capture the competing and epidemic spreading nature of a conflicting information pair, and identify two pivots in the lifetime of the undesired information, namely the *extinction time* τ_e and *half-life time* $\tau_{\frac{1}{2}}$, to quantify the dynamic evolution in the time domain.

We find both lifetime metrics are upper bounded by functions of $\eta(\mathcal{G})$, the Cheeger constant measuring the level of bottleneck-ness of $\mathcal{G}$, which indicates the lifetime of the undesired information does not always decrease with the network size n. When edges of the network become less and more concentrated such that $\eta(\mathcal{G}) = O(\log n)$, the lifetime of the virus will decrease with n, as exemplified by the dichotomy of $\mathbb{E}(\tau_e)$ and $\mathbb{E}(\tau_{\frac{1}{2}})$ for two extreme topologies, *i.e.*,

Considering $\eta(\mathcal{G})$ is difficult to obtain (NP-complete) for complex networks, we show that the lifetime of the undesired information can also be upper and lower bounded by functions of vertex eccentricities, which can be obtained with fast algorithms [112]. These bounds not only enable us to estimate the lifetime in large complex networks, but also imply a practical 'divide-and-conquer' guideline for information injection to be used as a countermeasure against viral spreading of undesired information.

We design an inference algorithm to estimate the number of information adopters at any time, given the initial condition and the topology of the network, which can be used to observe and predict the dynamic evolution with a finer resolution in time.

2.2 System Model

In this section, we introduce terminologies, assumptions and definitions of the Susceptible-Infected-Cured propagation model, which captures the competition between a pair of *conflicting* information, referred to as a *virus* and an *antidote*. Then based on the SIC model, we formally define *lifetime* metrics of the virus, so as to formulate the *when* problem. First, we specify the scope of conflicting information studied in this chapter.

2.2.1 Conflicting Information Pair: Virus x and Antidote a_x

Conflicting information has been defined as "two pieces of text that are extremely unlikely to be considered true simultaneously" in relevant information collection studies [36]. Considering that information takes various forms other than text, such as source code, operating status of devices, and commands in the motivating examples, we extend the definition of conflicting information to *mutually-exclusive* information that can not be possessed/admitted by the same individual at the same time. Particularly, we focus on a *virus-antidote* pair, in which the desired information (referred to as antidote a_x) is of dominant credibility/power over its undesired counterpart (referred to as virus x), such that it kills virus x if they are both present on the same vertex, but not vice versa. In other words, virus x can not re-infect an individual, who has admitted antidote a_x, which is different from the symmetric setting in existing models [94, 16], where virus x can re-infect an individual, who already has a copy of a_x, and a_x is treated as another virus symmetric to x, instead of an antidote to x.

The rationale behind the asymmetry in our model comes from observations in real-world examples and modeling accuracy concerns. First, for the case of security patch v.s. computer malware, and restoration commands v.s. (cascading) faulty status, the desired information (*e.g.*, security patch) is injected purposely by the system to eliminate the undesired information (*e.g.*, malware), and it is only reasonable that a malware can not leverage a fixed bug to attack the system. Second, for the case of clashing opinions and adoption of different products, once an individual is convinced(infected) by the newer and better product a_x, it is unlikely for him/her to switch back to the older product x, unless the older product has an upgrade (to the newest version) x'. This new injections of x' are considered as the beginning of a new epidemic process with x' as the antidote to virus a_x in our model. In the SI$_1$I$_2$S model [94, 16], x' and x are considered as the same virus competing with a_x, such that re-infection (or switching-back from a_x to x) is allowed, because of its focus on long-term (time $t \to \infty$) behaviors. A drawback of the existing approach is that x' and x are automatically associated with the same propagation speed. On the contrary, our model allows x, a_x, x' to have different propagation speeds, capturing competitions of both a_x v.s. x, and x' v.s. a_x, and is hence more accurate in modeling.

2.2.2 Network $\mathcal{G}(\mathcal{V}, \mathcal{E})$

The underlying *network* of information (virus x and antidote a_x) propagation is described as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where vertex set $\mathcal{V}$ corresponds to the set of individuals, such as devices in IoT, while edge set $\mathcal{E}$ corresponds to the set of connections between any of two individuals, such as the communication links between two devices. An edge $e(i, j)$ exists, if vertex i and j can directly exchange information. For any vertex $v \in \mathcal{V}$, its *neighborhood* $\mathcal{N}(v) := \{u \in \mathcal{V}|\ (u, v) \in \mathcal{E}\}$

is defined as the set of vertices that are directly connected to vertex v. The topology of the network can be described by its adjacency matrix $\mathbf{A}_{n\times n} = (a_{u,v})_{u,v\in\mathcal{V}}$, where $a_{u,v} = a_{v,u} = 1$, if there exits an edge $e(u, v) \in \mathcal{E}$.

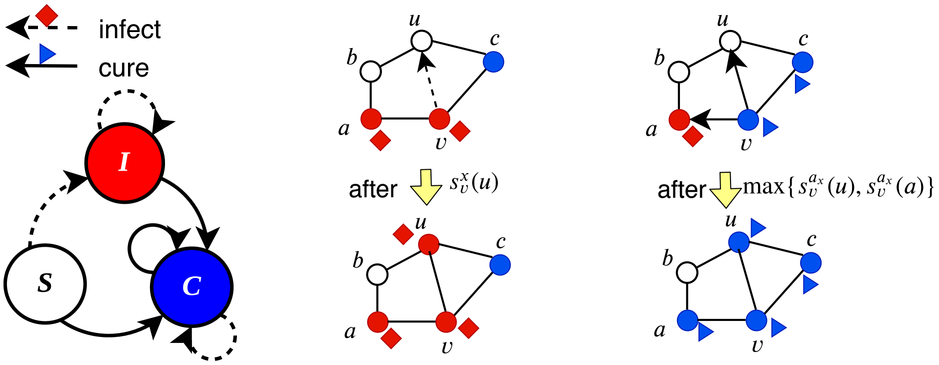
We make the following assumptions of network $\mathcal{G}$: i) It is undirected, that is, edge $e(i, j) = e(j, i)$ identifies mutual connections between vertices i and j. ii) It is connected, such that information (x and a_x) can spread to every vertex in $\mathcal{V}$. iii) It is static[1], that is, both size $n := |\mathcal{V}|$ and topology $\mathbf{A}$ of the network remain the same during the epidemic evolution.

2.2.3 Epidemic Propagation Process

Both virus x and antidote a_x spread in an *epidemic* manner on network $\mathcal{G}$. To describe this epidemic process, each vertex is associated with a *state* that can change over time.

2.2.3.1 State Transitions

Let r.v. $X_v^x(t) : \Omega \to \Lambda = \{0,\ 1,\ -1\}$ denote the state of vertex $v \in \mathcal{V}$ at time t. Values of X_v^x correspond to different nodal states, as shown in Figure 2.2a.



(a) State transitions. **(b)** An infection. **(c)** Curing events.

Figure 2.2 State transitions of individal vertices in the SIC propagation model.

Susceptible at time t: The default state $X_v^x(t) = 0$ (white circle with letter S in Figure 2.2a) indicates that neither the virus x nor the antidote a_x has reached vertex v by time t, so it is possible for v to be infected by x, or cured/immunized by a_x in the future, if any of them propagates to vertex v via contacts with other vertices.

Infected at time t_i: If a copy of virus x reaches susceptible vertex v at t_i, v becomes infected at t_i, which means $X_v^x(t_i) = 1$ and $\lim_{t\to t_i^-} X_v^x(t) = 0$. This *infect* action is shown by the dashed arrow from susceptible state to the *infected* state (red circle with letter I) in Figure 2.2a. At this state, vertex v will try to *infect*, *i.e.*, pass copies of virus x, indicated by red squares in

[1] Though the information dynamics is studied in a heterogeneous wireless network scenario such that the underlying connections are unstable, the social connection that directs information exchange is rather static, *i.e.*, less likely to change, during a data dissemination (a course of information propagation) process. Therefore, throughout this chapter, we assume the network $\mathcal{G}$ to be static.

Figure 2.2, to any u of its susceptible neighbors, $\mathcal{N}_S^x(t_i, v) = \{u \in \mathcal{N}(v) | X_u^x(t_i) = 0\}$, after a random period of time $s_v^x(u)$, as shown in Figure 2.2b. Vertex v will stay in infected state until it receives a copy of antidote a_x.

Cured at time t_c: If at t_c, a copy of antidote a_x reaches vertex v (solid arrows in Figure 2.2a) for the first, *i.e.*, $\lim_{t \to t_c^-} X_v^x(t) \geq 0$, the state of v changes to *cured* at t_c, that is, $X_v^x(t) = -1$, as shown as the blue circle with letter C in Figure 2.2. At this state, vertex v will pass copies of antidote (indicated by blue triangles) to any u of its neighbors $\mathcal{N}_{NC}^x(t, v) = \{u \in \mathcal{N}(v) | X_u^x(t_i) \geq 0\}$ after a random period of time $s_v^{a_x}(u)$, as shown in Figure 2.2c. Vertex v will stay cured for the rest of the time, *i.e.*, $X_v^x(t) = -1$ for any $t > t_c$.

2.2.3.2 Propagation Rules

As shown in Figure 2.2, the state transitions of vertex v are driven by an infection event of virus x, or a curing event of antidote a_x, whose speeds are controlled by the infection intervals $\{s_u^x(v)\}_{v \in \mathcal{N}_S(t,u)}$, and the curing intervals $\{s_u^{a_x}(v)\}_{v \in \mathcal{N}_{NC}(t,u)}$, respectively. These intervals can be of arbitrary lengths, but to make this problem tractable, we follow the convention in [94, 59], and assume time homogeneity for the propagation process: For any vertex u, random intervals $\{s_u^x(v)\}_{v \in \mathcal{N}_S(t,u)}$ and $\{s_u^{a_x}(v)\}_{v \in \mathcal{N}_{NC}(t,u)}$ are two groups of r.v.'s satisfying i) pairwise independent, and ii) exponentially distributed with parameters $\beta_{u,v}^x$ and $\gamma_{u,v}^x$, respectively.

From the perspective of time, $\beta_{u,v}^x = \beta_{v,u}^x$ is known as the *virulence* (or *infection rate*) of virus x, while $\gamma_{u,v}^x = \gamma_{v,u}^x$ as the *curing rate* of antidote a_x, representing how frequently a copy of virus x and antidote a_x is exchanged via edge $e(u, v)$, respectively. Their formal definition are given as

$$\beta_{u,v}^x := \lim_{t \to 0^+} \frac{\mathbb{P}(s_u^x(v) \leq t)}{t}, \tag{2.1}$$

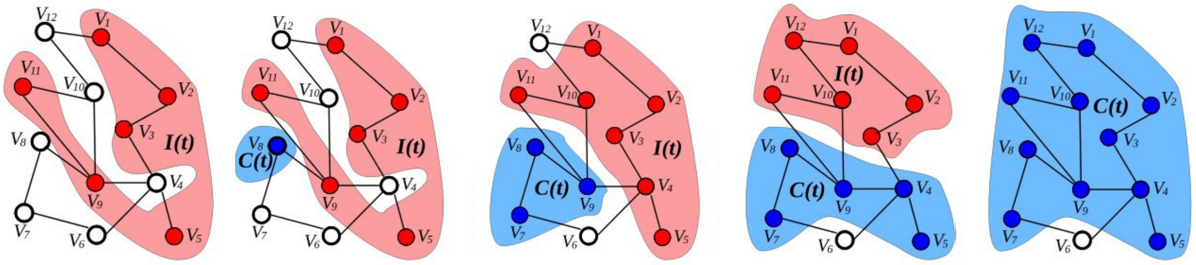
$$\gamma_{u,v}^x := \lim_{t \to 0^+} \frac{\mathbb{P}(s_u^{a_x}(v) \leq t)}{t}, \tag{2.2}$$

where $s_u^x(v)$ ($s_u^{a_x}(v)$, respectively) is the time period between the infection (curing) of u, and the time when u passes a copy of virus x (antidote a_x) to v vie edge $e(u, v)$.

From the perspective of probability, $\beta_{u,v}^x$ is also known as the infection probability over unit time on edge $e(u, v)$, which can be explained by considering a simple network composed of two connected vertices, *i.e.*, $\mathcal{V} = \{u, v\}$. In this case, at time t, given $X_v^x(t) = 0$, the probability that v gets infected by u in Δt is

$$\mathbb{P}\Big(X_v^x(t + \Delta t) = 1 | X_v^x(t) = 0\Big) = \Delta t \cdot \beta_{u,v}^x \cdot \mathbb{1}_{\{X_u^x(t)=1\}} + o(\Delta t), \tag{2.3}$$

where r.v. $s_u^x(v) \sim Exp(\beta_{u,v}^x)$, with mean $\mathbb{E}(s_u^x(v)) = \frac{1}{\beta_{u,v}^x}$. Similar results also apply to r.v. $s_u^{a_x}(v)$ and curing rate $\gamma_{u,v}^x$. Therefore, when the time interval is of unit length, *i.e.*, $\Delta t = 1$, the state transition probability $\mathbb{P}\Big(X_v^x(t + \Delta t) = 1 | X_v^x(t) = 0\Big) = \mathbb{P}(s_u^x(v) \leq t) = \beta_{u,v}^x$ equals to the infection rate $\beta_{u,v}^x$ in value, which bridges the gap between continuous-time modeling/analysis



(a) $t \to t_0^-$ **(b)** $t = t_0$ **(c)** $t = t_1$ **(d)** $t = t_2$ **(e)** $t = t_0 + \tau_e$

Figure 2.3 An example of an SIC epidemics on a network of 12 vertices.

and discrete-time simulations.

2.2.4 Problem Formulation

The SIC model in the previous subsection defines the propagation at individual vertex level. At the system level, the evolution of the SIC dynamics can be described by the set of vertices that are susceptible, infected, and cured, respectively, because at any time t, vertices set $\mathcal{V} = \mathcal{S}^x(t) \cup \mathcal{I}^x(t) \cup \mathcal{C}^x(t)$, where $\mathcal{S}^x(t)$, $\mathcal{I}^x(t)$, $\mathcal{C}^x(t)$ are mutually disjoint sets: the susceptible set $\mathcal{S}^x(t) := \{v \in \mathcal{V} : \ X_v^x(t) := 0\}$, the infected set$\mathcal{I}^x(t) = \{v \in \mathcal{V} : \ X_v^x(t) := 1\}$, and the cured set $\mathcal{C}^x(t) = \{v \in \mathcal{V} : \ X_v^x(t) = -1\}$. Evolution of an SIC epidemic process can then be captured by the time-varying *infection count*, $I^x(t) := |\mathcal{I}^x(t)|$, and the *cured count*, $C^x(t) := |\mathcal{C}^x(t)|$. For the ease of notation, we suppress x, and write $X_v(t)$, $\mathcal{S}(t)$, $\mathcal{I}(t)$, $\mathcal{C}(t)$ instead.

Figure 2.3 shows the evolution process of an SIC dynamics over a network with 12 vertices. As shown in Figure 2.3a, before the antidote a_x is injected, $I(t_0-) = 6$ vertices in $\mathcal{I}(t_0^-) = \{v_1, v_2, v_3, v_5, v_9, v_{11}\}$ are infected (colored in red). Then at t_0, one unit of antidote is given to vertex v_8, and cures it immediately (indicated as blue), that is, $\mathcal{C}(t_0) = \{v_8\}$, as shown in Figure 2.3b. As time t proceeds, states of the 12 vertices change as illustrated in Figure 2.3c-e. Eventually, the virus dies out at time $t = t_0 + \tau_e$, because $I(t) = 0$.

As shown in the example (Figure 2.3), the evolution of the SIC dynamics is recorded by the infection and cured counts across time. Based on these system-level states, we define lifetime metrics of the virus to answer *when* the undesired information dies out. Specifically, we identify two pivots in time, namely, the extinction time and half-life time, to quantify *how fast* the virus dies, which also indicate the effectiveness of an instance of antidote distribution.

Definition 2.1. *For an SIC dynamic of virus x and antidote a_x, the* ***extinction time*** *of virus x, denoted as τ_e, is defined as the time interval between t_0 and the infected set $\mathcal{I}(t)$ becomes empty for the first time, that is,*

$$\tau_e := \inf\{t > t_0 : \mathcal{I}(t) = \phi\} - t_0, \tag{2.4}$$

where t_0 is the time instant when C_0 copies of antidote a_x are injected into the network $\mathcal{G}$.

The extinction time τ_e is a finite[2] r.v. on measurable space $(\Omega^n, 2^{\Omega^n}, \mathbb{P})$. It answers *when the virus dies out*, because time instant $t_0 + \tau_e$ marks the *end point* of the virus's life in $\mathcal{G}$. In other words, the infection count decreases from $I(t_0)$ to 0 during a time interval of length τ_e, so we know the virus (undesired information) dies out at an average speed of $\frac{I(t_0)}{\tau_e}$. But it is still not clear how such speed changes during the lifetime of the virus, *i.e.*, *how fast* the virus dies, out, which answers a lot of realistic questions on the impact of the virus, *e.g.*, when will the majority of individuals be free of the undesired information? To answer this question, we identify another pivot in time, and define the half-life time of the virus.

Definition 2.2. *For an SIC dynamic of virus x and antidote a_x, the* ***half-life time*** *of the virus epidemic, denoted as $\tau_{\frac{1}{2}}$, is defined as the time interval between t_0 and the last time that event $\{I(t) \geq \frac{1}{2}I(t_0)\}$ happens after t_0, that is,*

$$\tau_{\frac{1}{2}} := \sup\{t \in [t_0, t_0 + \tau_e] : I(t) \geq \frac{1}{2}I(t_0)\} - t_0, \tag{2.5}$$

where $I(t_0) > 0$ is the initial infection count at t_0.

The half-life time $\tau_{\frac{1}{2}} : \Omega^n \to [t_0, t_0 + \tau_e]$ is also a finite r.v. on the same measurable space as r.v. τ_e. The term *half-life* is originally from Chemical Kinetics, which describes the decay of discrete entities. Unlike in Chemical Kinetics, where half-life is the mean, we define half-life as the actual time interval until event $\{I(t) \geq \frac{1}{2}I(t_0)\}$ happens for the *last* time. The physical meaning of $\tau_{\frac{1}{2}}$ can be explain as follows: the competition between the undesired information and its desired counterpart mainly takes place before pivot time $t_0 + \tau_{\frac{1}{2}}$, after which the undesired information can be viewed as controlled, because the number of its victim will never exceeds the threshold $I(t_0)/2$. In this sense, the two lifetime metrics, extinction time τ_e and half-life time $\tau_{\frac{1}{2}}$, illustrate *how fast* the virus dies, because for a fixed initial condition ($I(t_0)$ and $C(t_0)$), the larger the gap $\tau_e - \tau_{\frac{1}{2}}$, the faster the undesired information dies during its most hazardous phase $[t_0, t_0 + \tau_{\frac{1}{2}}]$.

Figure 2.4 illustrates the extinction time and the half-life time for the example of the SIC dynamics shown in Figure 2.3, where a red arrow corresponds to an infection, and a blue one represents a curing event. At $t_0 + \tau_{\frac{1}{2}}$, the infection count of the system drops to 3 ($= \frac{1}{2}I_0$), and never exceeds 3 again, which implies that the virus epidemic has been restricted to a limited area, or equivalently, under control. At $t_0 + \tau_e$, the virus dies out.

Without loss of generality, we let $t_0 = 0$, and denote $I(0)$ as I_0 (and $C(0) = C_0$) for the ease of notation. All the events we discuss hereafter take place in the observation window $[0, \tau_e]$. We further assume that both the infection rate β and curing rate γ are constant on every edge of the network, which is commonly adopted in information propagation studies.

Under the proposed SIC model, we restate our research question as follows: Consider a pair of conflicting information $(x,\ a_x)$, in which x is the virus with virulence β, and a_x is the antidote

[2] To be more accurate, r.v. τ_e is almost surely (a.s.) finite, that is, $\mathbb{P}(\tau_e < \infty) = 1$, when the network $\mathcal{G}$ is connected. The reason behind this is that it can be written as the summation of a finite number of exponential r.v.'s, each of which is a.s. finite.

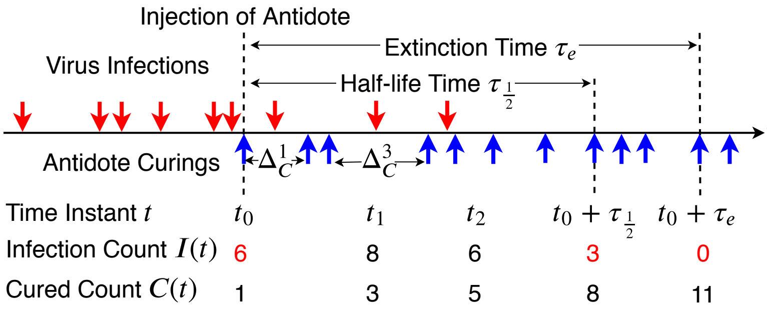


Figure 2.4 Illustration of the extinction time τ_e and half-life time $\tau_{\frac{1}{2}}$ of the virus under the SIC epidemics example in Figure 2.3: At time t_0, $C_0 = 1$ copy of antidote is injected into the network.

with curing rate γ. At time $t = 0$, C_0 copies of antidote are distributed in network $\mathcal{G}(\mathcal{V}, \mathcal{E})$, when the infection count equals to I_0.

- *When*: What is the expected extinction time $\mathbb{E}(\tau_e)$ and half-life time $\mathbb{E}(\tau_{\frac{1}{2}})$ of virus x?
- *Countermeasure*: How to properly select vertices in $\mathcal{C}_0$ to distribute antidotes, such that the lifetime of the virus ($\mathbb{E}(\tau_e)$ and $\mathbb{E}(\tau_{\frac{1}{2}})$) can be shortened?
- *Evolution*: Given time t, how to estimate the infection count $I(t)$ and cured count $C(t)$?

We answer these questions sequentially in the following sections. As mentioned in the introduction, the main obstacle of this study resides in the introduction of network $\mathcal{G}$, a structure with numerous topological properties, some of which are of particular importance to the epidemic spreading of information, as evidenced in [42, 59, 83]. Therefore, for the first question, which is the main objective of this chapter, we start from simple networks with two special topologies, to gain insights for solutions in networks with arbitrary topologies.

2.3 Lifetime of the Undesired Information in Networks with Simple Topologies

We first examine the lifetime of the virus in the complete graph K_n and the star graph S_n, which are not only common network topology themselves, but also essential components of complex networked systems.

2.3.1 Bounds for Complete Networks K_n

A complete graph K_n, is a fully connected graph with n vertices, in which for any pair of vertices $v_i, v_j \in V(K_n), i \neq j$, there exists an edge $e(i,j) \in E(K_n)$ between them, and hence $|\mathcal{E}(K_n)| = n(n-1)/2$. This topology is frequently seen in networks that require high reliability, *e.g.*, network of AS routers, or networks that are densely connected everywhere, *e.g.*, a household or a community, where every individual is familiar with one another. It has two key characteristics:

i) it is the most densely connected simple graph, because it has the maximum number of edges; ii) it is regular, because every vertex is inter-changable with another, which means they have the same vertex metrics, such as degree and centrality. In such networks, the expected lifetime metrics of the undesired information are upper bounded by the following theorem.

Theorem 2.1. *Consider an SIC epidemic in action on a complete network K_n, with curing rate γ, and initial condition $(I_0,\ C_0)$. The expected extinction time $\mathbb{E}(\tau_e)$ and half-life time $\mathbb{E}(\tau_{\frac{1}{2}})$ of the virus can be bounded above as*

$$\mathbb{E}(\tau_e) < \frac{1}{\gamma n}\left[2 + \ln\frac{(n-1)(n-C_0)}{C_0}\right], \tag{2.6}$$

$$\mathbb{E}(\tau_{\frac{1}{2}}) < \frac{1}{\gamma n}\left[2 + \ln\frac{(n-1)(n-1-\lceil I_0/2\rceil)}{\lceil I_0/2\rceil + 2}\right]; \tag{2.7}$$

when $C_0 \geq 2$, we also have

$$\mathbb{E}(\tau_e) < \frac{2}{\gamma(n-1+C_0)}\ln(n-C_0), \tag{2.8}$$

$$\mathbb{E}(\tau_{\frac{1}{2}}) \leq \frac{2}{\gamma(n-1-\lceil I_0/2\rceil + C_0)}\left(1 + \ln\frac{n-C_0}{\lceil I_0/2\rceil + 1}\right), \tag{2.9}$$

where $n = |\mathcal{V}|$ is the size of the complete network K_n.

Remark 2.1. *To prove this theorem, we first present a set of simple bounds on the exact value of the extinction time τ_e and the half-life time $\tau_{\frac{1}{2}}$, which is intuitive and only concerns intervals of curing events, as the starting step of the competing case in Theorem 2.1.*

Lemma 2.1. *Consider a network $\mathcal{G}(\mathcal{V},\mathcal{E})$ with $|\mathcal{V}| = n$ under SIC dynamics. At $t = 0$, there are initially I_0 infected nodes and C_0 units of antidote disseminated. We have the following bounds for the extinction time and the half-life time.*

$$\sum_{k=1}^{I_0}\Delta_C^k \leq \tau_e \leq \sum_{k=1}^{n-C_0-1}\Delta_C^k, \tag{2.10}$$

$$\sum_{k=1}^{I_0/2}\Delta_C^k \leq \tau_{\frac{1}{2}} \leq \sum_{k=1}^{n-C_0-1-\lceil I_0/2\rceil}\Delta_C^k, \tag{2.11}$$

where Δ_C^k denotes the interval between the (C_0+k)-th and (C_0+k-1)-th curings in the network, as shown in Figure 2.4.

Proof. Note that all the initial C_0 curings happen at $t = 0$. Recall our definition of time interval between curings $\Delta_C^k = \tau_C^{k+C_0} - \tau_C^{k+C_0-1}$. At time τ_C^{k-1}, the number of cured vertices can be calculated by $C\big(\tau_C^{k+C_0-1}\big) = k - 1 + C_0$.

The upper bounds come from asymmetric immunity, that is, a cured vertex will never be infected again. So number of cured vertices $C(t)$ is non-decreasing, then half-life of the virus

$t_{\frac{1}{2}}$ is bounded above by the spreading time of antidote epidemic. At $t_1 = inf\{t > 0|C(t) = n - \lceil I_0/2 \rceil\}$, $I(t_1) + S(t_1) = n - C(t_1) = \lceil I_0/2 \rceil$, therefore $I(t) \leq \lceil I_0/2 \rceil, \forall t \geq t_1$.

For the lower bounds, τ_e is greater than or equal to the curing time of the initially infected set $\mathcal{I}(0)$, because as the antidote spread, the virus may have collected new victims already. Now consider a 'smart' antidote with the same curing rate γ, or equivalently, let the cured vertices spread the antidote in a 'smart' way, in which it always cures its infected neighbors first. For each realization of the SIC epidemic, this 'smart' antidote distribution is equivalent to re-arranging the sequence of curing events in an SIC epidemic. Similar statement is also true for the half-life time. □

Proof. **(Theorem 2.3.)** The process of cured count, $\{C(t)\}_t$ is a continuous time Markov chain with transition rate

$$c \rightarrow c+1 \text{ at rate } \gamma c(n-c) \quad \forall\ C_0 \leq c \leq n-1,$$

and hence $\mathbb{E}(\Delta_C^c) = \frac{1}{\gamma c(n-c)} \quad \forall\ C_0 \leq c \leq n-1$, because r.v. Δ_C^c is Exponentially distributed with parameter $\gamma c(n-c)$.

Then by Lemma 2.1, we have

$$\begin{aligned}\mathbb{E}(\tau_e) &\leq \mathbb{E}\Big(\sum_{c=C_0}^{n-1} \Delta_C^c\Big) = \frac{1}{\gamma n}\sum_{c=C_0}^{n-1}(\frac{1}{c} + \frac{1}{n-c}) \\ &\leq \frac{1}{\gamma n}(\mathcal{H}_{n-1} - \mathcal{H}_{C_0-1} + \mathcal{H}_{n-C_0}) \\ &< \frac{1}{\gamma n}\big[2 + \ln\frac{(n-1)(n-C_0)}{C_0}\big],\end{aligned}$$

where $\mathcal{H}_n$ is the *Harmonic Number*, and $\ln(n+1) < \mathcal{H}_n \leq \ln(n)+1$. Similarly, the same method can be used to derive upper bound of half-life for clique.

$$\begin{aligned}\mathbb{E}(\tau_{\frac{1}{2}}) &\leq \mathbb{E}\left(\sum_{c=C_0}^{n-1-\lceil I_0/2 \rceil} \Delta_C^c\right) = \sum_{c=C_0}^{n-1-\lceil I_0/2 \rceil} \frac{1}{\gamma c(n-c)} \\ &\leq \frac{1}{\gamma n}(\mathcal{H}_{n-1-\lceil I_0/2 \rceil} - \mathcal{H}_{\lceil I_0/2 \rceil+1} + \mathcal{H}_{n-1}) \\ &< \frac{1}{\gamma n}\big[2 + \ln\frac{(n-1)(n-1-\lceil I_0/2 \rceil)}{\lceil I_0/2 \rceil + 2}\big].\end{aligned}$$

Similarly technique can be applied for the case of $C_0 \geq 2$. □

Theorem 2.1 implies that both the expected extinction time and half-life time decreases with size n of the clique as $O(\frac{\log n}{n})$, which means the larger the network, the quicker the undesired information dies, given the same initial condition of the dynamics. The reason behind this is that, dense connections among vertices make it difficult for the virus to dodge contact with antidotes. In other words, even though both virus and antidote can spread faster, due to the larger number of edges, dense connections work in favor of the antidote propagation. On the other hand, with respect to the severeness of virus infection upon antidote injection, we can see the half-life time decreases with the initial infection count I_0 as $O(\log \frac{A}{I_0})$, where A is a function of n and C_0 that does not vary with I_0. In this case (larger I_0), the larger gap $\tau_e - \tau_{\frac{1}{2}}$ indicates that, competition between the antidote and the virus is only fierce for a short period of time, and the larger portion of extinction time is spent on extinguishing cornered virus infections.

From the perspective of connectivity, any other simple network topology of the same size n has strictly less edges than the complete graph K_n, as a result of which the undesired information (virus) will die slower with high probability in these networks, even with the same initial condition (I_0 and C_0). Bases on this observation, we also have the following lower bounds as a corollary of Theorem 2.1.

Corollary 2.1. *For an SIC epidemic with curing rate γ on an arbitrary network $\mathcal{G}$ of size n, the expected extinction time and half-life time of the virus can be bounded below as*

$$\mathbb{E}(\tau_e) \geq \frac{1}{\gamma n}\big[\mathcal{H}_{C_0+I_0-1} - \mathcal{H}_{C_0-1} + \mathcal{H}_{n-C_0} - \mathcal{H}_{n-C_0-I_0}\big], \tag{2.12}$$

$$\mathbb{E}(\tau_{\frac{1}{2}}) \geq \frac{1}{\gamma n}\big[\mathcal{H}_{C_0+\lceil I_0/2\rceil-1} - \mathcal{H}_{C_0-1} + \mathcal{H}_{n-C_0} - \mathcal{H}_{n-C_0-\lceil I_0/2\rceil}\big], \tag{2.13}$$

where $\mathcal{H}_k = \sum_{j=1}^{k} \frac{1}{j}$ is the k-th Harmonic Number.

Proof. Let Δ_C^k denote the time interval between the $(C_0 + k)$-th and the $(C_0 + k + 1)$-th curing event, as shown in Figure 2.4. The expected extinction time and half-life time can be lower bounded by the time that the $C_0 + I_0$-th and $C_0 + \lfloor I_0/2 \rfloor$-th vertex first receives a copy of the antidote respectively, which are shown in Lemma 2.1. On the other hand, the spreading time of the antidote is statistically bounded below by that in a complete network, *i.e.*, when $\mathcal{G} = K_n$. In this case, the value of an inter-curing interval follows an Exponential distribution, that is, $\Delta_C^k \sim Exp(\gamma(C_0 + k - 1)(n - k - C_0 + 1))$, because it is the minimum r.v. out of $n - C_0 - k$ exponentially distributed r.v.'s. Therefore,

$$\mathbb{E}(\tau_e) \geq \mathbb{E}\left(\sum_{k=1}^{I_0} \Delta_C^k\right) = \sum_{k=1}^{I_0} \mathbb{E}(\Delta_C^k) = \frac{1}{\gamma n}\sum_{k=1}^{I_0}\left[\frac{1}{C_0 + k - 1} + \frac{1}{n - C_0 - k + 1}\right],$$

and Eq. (2.12) follows from the definition of Harmonic number. Proof of the lower bound of the expected half-life time $\mathbb{E}(\tau_{\frac{1}{2}})$ is similar and hence omitted. □

2.3.2 Bounds for Star Networks S_n

As opposed to the most densely connected complete network K_n, a star network S_n is composed of a hub v_1 and $n - 1$ leaves (peripheral vertices), such that every piece of information from a leaf vertex has to go through the hub to reach another vertex. In other words, edges only exists between the hub and leaves, creating a huge bottle-neck in the middle, which can be viewed as a high-degree of heterogeneity among vertices. The star topology is also of great importance, due to its natural link to artificial structures, *e.g.*, a WiFi access network within the coverage of an access point, and the ego-network of a high degree node in OSNs.

Consider same SIC dynamic with infection rate β and curing rate γ on a star S_n. In an extreme case, if a copy of antidote is first given to the hub, then infection count $I(t)$ will be monotonically decreasing with time t, which is less interesting, because it is impossible for the virus to claim new victims, and hence no competition. Therefore, we consider the case that antidotes are distributed to peripheral vertices at time 0 in the following theorem.

Theorem 2.2. *For an SIC epidemic over a start network in which the initial infection count satisfies $I_0 \geq 2$, the expected extinction time $\mathbb{E}(\tau_e)$ and half-life $\mathbb{E}(\tau_{\frac{1}{2}})$ follow upper bounds:*

$$\mathbb{E}(\tau_e) < \frac{1}{\gamma}\left[\frac{1}{C_0} + 1 + \ln(I_0 - 1) + \frac{\beta}{\gamma C_0 (I_0 - 1)}\right], \tag{2.14}$$

$$\mathbb{E}(\tau_{\frac{1}{2}}) < \frac{1}{\gamma}\left[\frac{1}{C_0} + 1 + \ln(\frac{I_0 - 1}{\lceil I_0/2 \rceil}) + \frac{\beta}{\gamma C_0 (I_0 - 1)}\right]. \tag{2.15}$$

Proof. Let r.v. τ_0 denote the time when the hub is cured, then $I(t)$ is non-increasing after τ_0. Let Δ denote the number of new infections until τ_0, T_k denote the time that the cured hub disseminates antidote to the k infected peripheral vertices. Then T_k is the k-th minimum out of the $I_0+\Delta-1$ i.i.d. Exponential r.v.'s with parameter γ. If $k = I_0+\Delta-1$, T_k will be the maximum of k i.i.d Exponential r.v.'s with parameter γ, or equivalently $\mathbb{E}(T_{I_0+\Delta-1}) = \frac{1}{\gamma}\mathcal{H}_{I_0+\Delta-1}$. Then we can write the expectation of extinction time τ_e, with $\mathbb{E}(\tau_e) = \mathbb{E}(\tau_0) + \mathbb{E}(T_{\Delta+I_0-1})$, where τ_0 satisfies Exponential distribution with parameter $(C_0\gamma)$. Hence, its probability density function $f_{\tau_0}(t) = C_0\gamma e^{-C_0\gamma t}$, and its mean $\mathbb{E}(\tau_0) = \frac{1}{C_0\gamma}$. So

$$\begin{aligned}\mathbb{E}(T_{\Delta+I_0-1}) = \mathbb{E}\big(\mathbb{E}(T_{\Delta+I_0-1}|\Delta)\big) &= \frac{1}{\gamma}\sum_{k=0}^{n-C_0-I_0}\mathcal{H}_{k+I_0-1}\cdot\mathbb{P}(\Delta = k)\\ &< \frac{1}{\gamma}\sum_{k=0}^{n-C_0-I_0}\Big(1 + \ln(k + I_0 - 1)\Big)\cdot\mathbb{P}(\Delta = k).\end{aligned}$$

Let function $g(k) = \ln(k + I_0 - 1)$. Notice that $g(\cdot)$ is concave in $[1, n - C_0 - I_0]$. Then

$$\begin{aligned}\mathbb{E}(T_{\Delta+I_0-1}) < \frac{1}{\gamma} + \frac{1}{\gamma}\mathbb{E}\big(g(\Delta)\big) = \frac{1}{\gamma} + \frac{1}{\gamma}\mathbb{E}\big(\mathbb{E}(g(\Delta)|\tau_0)\big) &= \frac{1}{\gamma} + \frac{1}{\gamma}\int_0^\infty \mathbb{E}\big(g(\Delta)|\tau_0 = t\big)\cdot f_{\tau_0}(t)dt\\ &\overset{Jensen}{\leq} \frac{1}{\gamma} + \frac{1}{\gamma}\int_0^\infty g\big(\mathbb{E}(\Delta|\tau_0 = t)\big)\cdot f_{\tau_0}(t)dt.\end{aligned}$$

Now consider r.v. $\mathbb{E}(\Delta|\tau_0 = t)$. Since the network S_n is a star, and Δ is the number of new infections until time instant τ_0 when the hub gets cured, these Δ new infections of different susceptible vertices are mutually independent. Therefore, given a fixed time instant $\tau_0 = t$, r.v.

Δ obeys Binomial distribution $B(n - C_0 - I_0, 1 - e^{-\beta t})$, *i.e.*,

$$\begin{aligned}\mathbb{P}(\Delta = k|\tau_0 = t) &= \mathbb{P}(k \text{ out of } n - C_0 - I_0 \text{ vertices are infected before } t)\\ &= \binom{n - C_0 - I_0}{k}(e^{-\beta t})^{n-C_0-I_0-k}(1 - e^{-\beta t})^k.\end{aligned}$$

Consequently, we have the conditional expectation $\mathbb{E}(\Delta|\tau_0 = t) = (n - C_0 - I_0) \cdot (1 - e^{-\beta t})$ as a function of t, which takes value in $[0, \tau_0]$. Note time instant τ_0 is also a r.v., whose range is $[0, \infty)$, so integrating over this range we have the expected time until the $\Delta + I_0 - 1$-th curing as

$$\begin{aligned}\mathbb{E}(T_{\Delta+I_0-1}) &\leq \frac{1}{\gamma} + \frac{1}{\gamma}\int_0^\infty g\big((n - C_0 - I_0) \cdot (1 - e^{-\beta t})\big) \cdot f_{\tau_0}(t)dt\\ &= \frac{1}{\gamma} - \frac{1}{\gamma}\int_0^\infty \ln\Big[(n - C_0 - 1) - (n - C_0 - I_0)e^{-\beta t}\Big]d\big(e^{-C_0\gamma t}\big)\\ &= \frac{1}{\gamma}\big[1 + \ln(I_0 - 1)\big] + \frac{\beta(n - C_0 - I_0)}{\gamma(n - C_0 - 1)} \cdot \int_0^\infty \frac{e^{-C_0\gamma t}}{e^{\beta t} - \frac{n-C_0-I_0}{n-C_0-1}}dt\\ &= \frac{1}{\gamma}\big[1 + \ln(I_0 - 1)\big] + \frac{\beta}{\gamma}\sum_{k=1}^{\infty}\frac{1}{C_0\gamma + k\beta} \cdot \Big(\frac{n - C_0 - I_0}{n - C_0 - 1}\Big)^k\\ &\leq \frac{1}{\gamma}\big[1 + \ln(I_0 - 1)\big] + \frac{\beta}{\gamma^2 C_0(I_0 - 1)},\end{aligned}$$

which is the upper bound of $\mathbb{E}(\tau_e)$ in Eq. (2.14). Similarly, when considering the half-life time, at τ_0, there are $\Delta + I_0 - 1$ infected peripheral vertices. Thus $T_{\Delta+\lceil I_0/2\rceil-1} = t$ indicates the time when there are only $\lceil I_0/2\rceil$ infected vertices left in the peripheral area, or equivalently, $\Delta + \lceil I_0/2\rceil - 1$ out of $\Delta + I_0 - 1$ i.i.d Exponential r.v's are less than t. Therefore,

$$\begin{aligned}\mathbb{E}(T_{\Delta+\lceil I_0/2\rceil-1}) &= \frac{1}{\gamma}(\mathcal{H}_{\Delta+I_0-1} - \mathcal{H}_{\lceil I_0/2\rceil-1})\\ &< 1 + \ln(\Delta + I_0 - 1) - \ln(\lceil I_0/2\rceil - 1 + 1) = 1 + \ln\frac{\Delta + I_0 - 1}{\lceil I_0/2\rceil}.\end{aligned}$$

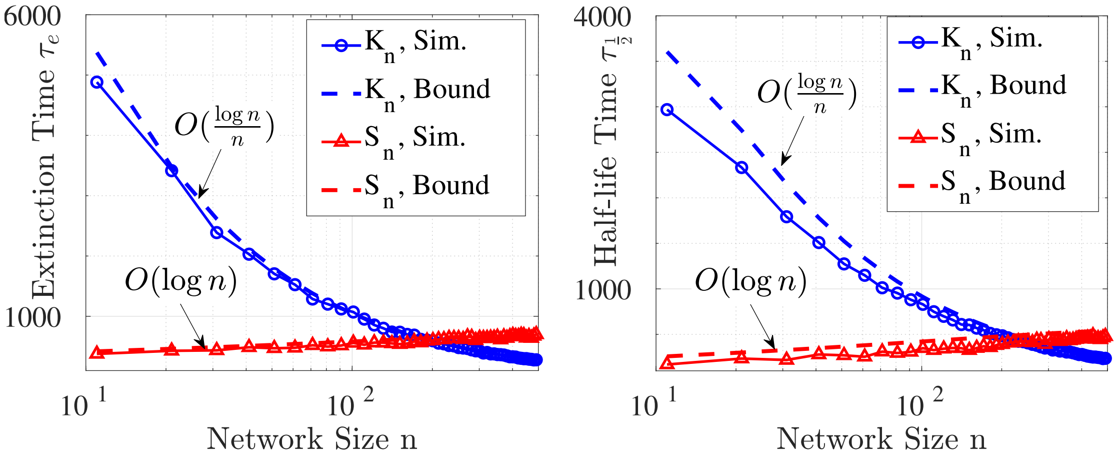
Let $g'(k) = \ln\frac{k+I_0-1}{\lceil I_0/2\rceil}$. Then we have the upper bound in Eq. (2.15) because

$$\mathbb{E}(\tau_{\frac{1}{2}}) = \mathbb{E}(\tau_0) + \mathbb{E}(T_{\Delta+\lceil I_0/2\rceil-1}) \overset{Jensen}{<} \frac{1}{C_0\gamma} + \frac{1}{\gamma} + \frac{1}{\gamma}\int_0^\infty g'\Big((n - C_0 - I_0) \cdot (1 - e^{-\beta t})\Big) \cdot f_{\tau_0}(t)dt.$$

□

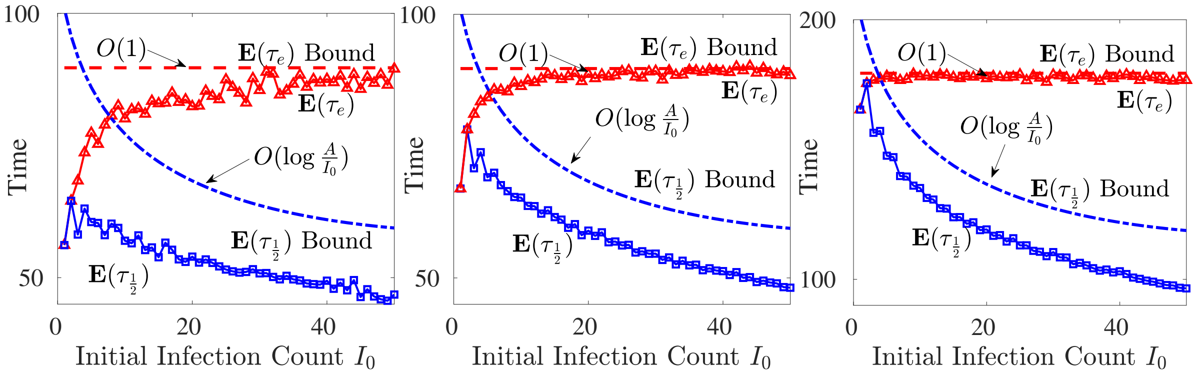
2.3.3 Numerical Simulation and Discussion

To validate upper bounds (dashed lines) in Theorem 2.1 and Theorem 2.2, numerical results (solid lines with markers) with respect to network size n and the initial infection count I_0 are shown in Figure 2.5 and Figure 2.6-2.7 respectively. As can be seen from all three figures, trends



(a) Extinction time $\mathbb{E}(\tau_e)$. **(b)** Half-life time $\mathbb{E}(\tau_{\frac{1}{2}})$.

Figure 2.5 Expected extinction time $\mathbb{E}(\tau_e)$ and half-life time $\mathbb{E}(\tau_{\frac{1}{2}})$ of an SIC epidemics ($C_0 = 1$, $I_0 = 10$), with respect to the network size n. In this simulation, we set the infection(β) and curing (γ) rates as $\beta = \gamma = 0.01$ in the star network S_n, and $\beta = \gamma = 0.0001$ in the complete network K_n, for a clearer comparison between the two topologies.



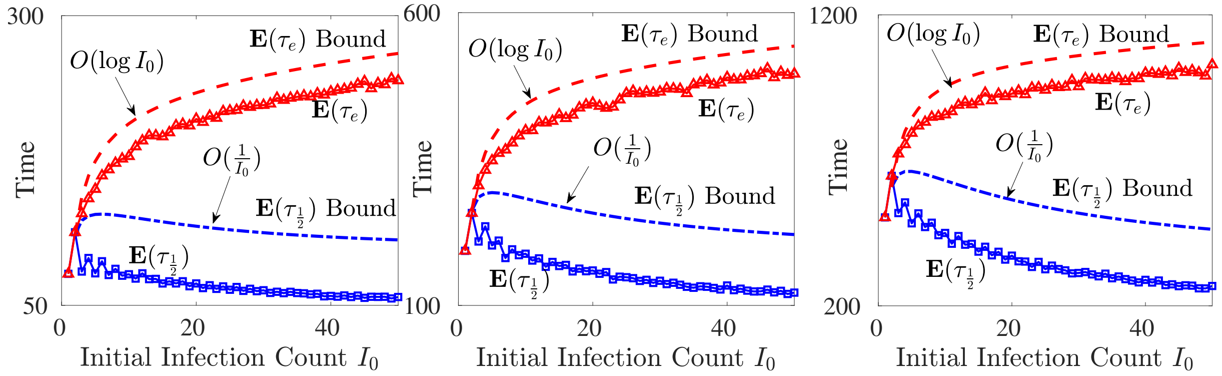
(a) $\beta = 0.0005,\ \gamma = 0.001,\ C_0 = 3$. **(b)** $\beta = \gamma = 0.001,\ C_0 = 3$. **(c)** $\beta = 0.001,\ \gamma = 0.0005\ C_0 = 3$.

Figure 2.6 Extinction time $\mathbb{E}(\tau_e)$ and Half-life time $\mathbb{E}(\tau_{\frac{1}{2}})$ in the complete graph K_{100}.

of both the expected extinction time and half-life time are well captured by the derived bounds. Comparing results for the two simple topologies, we highlight the following observations.

(1) There is a clear dichotomy in lifetime of the virus with respect to network size n, that is, virus dies faster in a larger complete network K_n, but slower in a larger star network S_n, as shown in the semi-log plots of Figure 2.5. Both $\mathbb{E}(\tau_e)$ and $\mathbb{E}(\tau_{\frac{1}{2}})$ are $O(\frac{\log n}{n})$ for complete networks (decreasing blue lines), while both are $O(\log n)$ for star networks (increasing red lines), as indicated by Theorem 2.1 and Theorem 2.2, respectively. This implies an interesting change in the propagation process, when a hub (bottleneck) emerges in the network.

(2) The infection/curing rates, β and γ, have the same effect on both metrics regardless of network topology, which can be observed by comparing sub-figures in Figure 2.6 and 2.7. To be more specific, they only lengthen or shorten time intervals between events, but do not change



(a) $\beta = 0.01,\ \gamma = 0.02,\ C_0 = 3.$ **(b)** $\beta = \gamma = 0.01,\ C_0 = 3.$ **(c)** $\beta = 0.01,\ \gamma = 0.005\ C_0 = 3.$

Figure 2.7 Extinction time $\mathbb{E}(\tau_e)$ and Half-life time $\mathbb{E}(\tau_{\frac{1}{2}})$ in the star network S_{100}.

the order of occurrences, resulting in a homogeneous scaling effect on both lifetime metrics.

(3) Impact of the initial infection count I_0 on the lifetime of the virus is shown in Figure 2.6 and 2.7, for complete network K_{100} and star network S_{100}, respectively. The mild zig-zag pattern of the half-life time (left end of all blue solid lines with markers) is caused by rounding-off the threshold $\lfloor \frac{1}{I_0} \rfloor$ when I_0 is small. Though the general trend is the same in both networks, *i.e.*, extinction time increases with I_0, while half-life time decreases with I_0, impact of topology is still apparent, as shown by the different scaling behavior over I_0 (marked in Figure 2.6 and 2.7).

As two extremes of network topology, the complete graph is regular (every vertex has the same degree) and most densely connected ($|\mathcal{E}| = \frac{n(n-1)}{2}$), while the star graph is highly irregular (due to the existence of the central hub) and sparse ($|\mathcal{E}| = n-1$). Comparing the two, we observe when edges concentrate and form a bottleneck in the network, the impact of network size n and initial condition I_0 changes drastically. Though results in these two simple networks do not directly apply to general cases, they shed lights on studying conflicting information propagation in general networks, by recognizing the importance of bottlenecks in networks.

2.4 Lifetime of the Undesired Information in Networks with Arbitrary Topologies

Considering the broad application scenarios of conflicting information propagation, it is likely that the underlying network $\mathcal{G}$ does not have nice topological properties like regularity, and is hence a *complex network* with unique topologies. To study *when* and *how fast* the undesired information dies out in such networks, we examine graph metrics, namely the Cheeger constant $\eta(G)$ and vertex eccentricities $\{\epsilon(v)\}_{v\in\mathcal{V}}$, to quantitatively link topological properties to the lifetime metrics ($\mathbb{E}(\tau_e)$ and $\mathbb{E}(\tau_{\frac{1}{2}})$) of the virus in an SIC epidemics through upper bounds.

2.4.1 Bounds by Considering the Edge-expansion Property

Recall in the comparison between complete networks and star networks, we find that the key to propagation behavior change is the existence of the central hub in star S_n, which forms a bottleneck in the network. Therefore, we first consider the metric that quantitatively measures the level of 'bottleneckness' of a network $\mathcal{G}$, the Cheeger constant $\eta(\mathcal{G})$, which is defined as $\eta(\mathcal{G}) := \inf\limits_{S\subset\mathcal{V},|S|\leq n/2} \frac{|\delta(S)|}{|S|}$, where $\delta(S) := [S, \mathcal{V}\setminus S]$ is the edge-cut of vertex set S, that is, the set of boundary edges between set S and its compliment set $\mathcal{V}\setminus S$.

As a graph expansion property [51], $\eta(\mathcal{G})$ identifies the 'narrowest' part of network $\mathcal{G}$, *i.e.*, the minimum boundary edges for as many as vertices. Intuitively, the larger the boundary set, the more difficult it is to break the network into isolated components by disconnecting edges, so $\eta(\mathcal{G})$ and other expansion properties are viewed as indicators of robustness, and hence studied in many applications [51]. Due to this property, it has been shown that $\eta(\mathcal{G})$ is of key importance to the spreading behavior of an epidemic process, under different propagation models, including SI [59, 141], SIS [59], and SIR [42] models. For the proposed SIC propagation model, in which *two* epidemic processes compete in finite time, we show in the following theorem that both the expected extinction time and half-life time are also bounded by $\eta(\mathcal{G})$, as $O(\frac{\log n}{\gamma\eta(\mathcal{G})})$.

Theorem 2.3. *For a network $\mathcal{G}$ with Cheeger constant $\eta(\mathcal{G})$, given the initial cured count C_0, the extinction time of the virus in an SIC epidemic with curing rate γ can be bounded above by*

$$\mathbb{E}(\tau_e) \leq \frac{1}{\gamma\eta(\mathcal{G})}[a\ln(n+b)+c], \tag{2.16}$$

1. *If $1 \leq C_0 < n/2$, $b = 0$, and $a = \frac{2}{\ln 4(C_0-1)}$, $c = -\big[\frac{4C_0-5}{8(C_0-1)^2} + \gamma_E\big]$.*
   Particularly when $C_0 = 1$, $a = \frac{2}{ln2}$, $c = 2\gamma_E$;

2. *If $C_0 \geq n/2$, $a = 1$, $b = -C_0 + 1$, $c = \frac{1}{2(n-C_0+1)} + \gamma_E$,*

where $\gamma_E \simeq 0.577$ is the Euler-Mascheroni constant.

When the initial infection count I_0 is given, the expected half-life time can be upper bounded[3] *when $2 \leq C_0 < n/2$,*

$$\mathbb{E}(\tau_{\frac{1}{2}}) \leq \frac{1}{\gamma\eta(\mathcal{G})}\left[\ln\frac{n^2}{2(C_0-1)I_0} - \frac{4C_0-5}{8(C_0-1)^2} + \frac{2I_0+1}{I_0^2}\right], \tag{2.17}$$

and when $C_0 = 1$, $\mathbb{E}(\tau_{\frac{1}{2}}) \leq \frac{1}{\gamma\eta(\mathcal{G})}\left[\ln\frac{n^2}{2I_0} + \frac{2I_0+1}{I_0^2}\right]$.

Proof. Let $\mathcal{V}_k := \{V|V \subset \mathcal{V}, |V| = k\}$ be the set of all vertex sets containing k vertices from the network $\mathcal{G}(\mathcal{V},\mathcal{E})$. Let $\mathcal{C}_k \in \mathcal{V}_k$ be the set of cured k vertices when $C(t) = k$. Let $T_k := \inf\{t|C(t) = k\}$ be the time that the cured count $C(t)$ reaches k. Note that $C(t)$ is

[3] For the case of $C_0 > n/2$, it is with high probability that $\tau_{\frac{1}{2}} = 0$, which is a less interesting case and hence not discussed.

monotonically increasing, and $C(0) = C_0$, $T_{C_0} = 0$. For any $k \geq C_0$,

$$\mathbb{E}(T_{k+1} - T_k) = \sum_{A \subset \mathcal{V}_k} \mathbb{E}\Big(T_{k+1} - T_k | \mathcal{C}(t) = A) \mathbb{P}(\mathcal{C}(t) = A\Big) = \mathbb{E}\left(\frac{1}{\gamma |\delta(\mathcal{C}_k)|}\right),$$

where $\delta(\mathcal{C}_k) := [\mathcal{C}_k, \mathcal{V} \setminus \mathcal{C}_k]$ is the edge cut of set $\mathcal{C}_k$, and

$$|\delta(\mathcal{C}_k)| \geq \begin{cases} \eta(\mathcal{G})k, & k < \frac{n}{2}, \\ \eta(\mathcal{G})(n-k), & k \geq \frac{n}{2}. \end{cases}$$

Also, by Lemma 2.1, we have $\mathbb{E}(\tau_e) \leq \sum_{k=C_0}^{n-1} \mathbb{E}(T_{k+1} - T_k)$. So when $2 \leq C_0 < \frac{n}{2}$, the expected extinction time can be upper bounded by

$$\begin{aligned} \mathbb{E}(\tau_e) &\leq \sum_{k=C_0}^{n/2} \frac{1}{\gamma \eta(\mathcal{G}) k} + \sum_{k=\frac{n}{2}+1}^{n-1} \frac{1}{\gamma \eta(\mathcal{G})(n-k)} = \frac{1}{\gamma \eta(\mathcal{G})} \left(2\mathcal{H}_{\frac{n}{2}} - \frac{2}{n} - \mathcal{H}_{C_0 - 1}\right) \\ &\overset{Franel}{<} \frac{1}{\gamma \eta(\mathcal{G})} \left[\ln \frac{n^2}{4(C_0 - 1)} - \frac{4C_0 - 5}{8(C_0 - 1)^2} - \gamma_E\right], \end{aligned}$$

where the last line follows from Franel's Inequality[98]. Similarly, the expected half-life time can be bounded by

$$\mathbb{E}(\tau_{\frac{1}{2}}) \leq \sum_{k=C_0}^{n/2} \frac{1}{\gamma \eta(\mathcal{G}) k} + \sum_{k=\frac{n}{2}+1}^{n-1-\lceil I_0/2 \rceil} \frac{1}{\gamma \eta(\mathcal{G})(n-k)}.$$

As for $C_0 \geq \frac{n}{2}$, the upper bound can be obtained by considering $\mathbb{E}(\tau_e) \leq \sum_{k=C_0}^{n-1} \frac{1}{\gamma \eta(\mathcal{G})(n-k)}$. □

Though not a sharp bound, Theorem 2.3 confirms the importance of the edge-expansion property, *i.e.*, the *Cheeger* constant $\eta(\mathcal{G})$, in epidemic spreading of information, particularly two pieces of conflicting information studied in this chapter. As can be seen, the more 'expandable' (strong connectivity with less vertices, towards the clique topology), the less time it takes to eliminate the virus, which explains the shorter rumor circulation time [131] than before, as OSNs (and Internet at large) become more connected nowadays. Particularly for the two simple network topologies, clique and star, studied in the previous section, Theorem 2.3 clearly explains the dichotomy of the scaling laws over the network size n: i) $\eta(K_n) = n/2 = O(n)$, which implies the $O(\frac{\log n}{n})$ scaling for the complete network K_n; while ii) $\eta(S_n) = O(1)$ for the star network S_n, and hence the $O(\log n)$ increasing over network size n.

For some large networks with special topological properties [51], existing results on its edge expansion properties allow us to estimate the lifetime of the undesired information in such networks as the network grow in size. For instance, Krishnasamy et.al. showed that for Erdös-Rényi random graph (ER) $G(n, p)$ with $p > \frac{32 \log n}{4}$, there is a high probability (over $1 - \frac{1}{n^2}$) that $\eta(G) \geq \frac{np}{4}$ [59, Corollary 2]. Applying Theorem 2.3, we know that the undesired information dies out in $O(\frac{logn}{\gamma n})$ time with high probability.

For a complex network with arbitrary topology, however, obtaining the Cheeger constant is a well-known hard problem (NP-complete [74]), especially when the network is large. In this case, $\eta(\mathcal{G})$ can be bounded by the *Cheeger Inequality* [29, 51], $\frac{\lambda_1}{2} \leq \eta(\mathcal{G}) \leq 2\sqrt{\lambda_1}$, where λ_1 is the second smallest eigenvalue of the graph Laplacian[4] of network $\mathcal{G}$. As we will show in the simulation later, $\frac{\lambda_1}{2}$ does not lead to a tight bound of the extinction time, so next we consider properties that are more accessible for general large networks.

2.4.2 Bounds by Considering Vertex Eccentricity

Recall in Theorem 2.2, after the hub of a star network is cured, the extinction time of the virus is bounded by the last curing event of an infected peripheral vertex, which depends on the longest time it takes a copy of antidote to pass though multiple hub-peripheral paths of length 1. In a general network $\mathcal{G}$, if we view every initially cured vertex $v \in \mathcal{C}(0)$ as a 'hub', the extinction time also depends on such hub-peripheral paths, whose lengths are measured in hop-count distances $\mathrm{dist}(\cdot,\cdot)$. In practice, distance between two vertices is easy to obtain from the adjacency matrix of a graph, and is therefore widely used in various applications, such as routing and influential node detection. Based on this graph metric, the *eccentricity* of a vertex v in network $\mathcal{G}$ is defined as the longest distance of between v and any other vertex in $\mathcal{V}$, that is, $\epsilon_{\mathcal{G}}(v) := \max_{u\in\mathcal{V}} dist_{\mathcal{G}}(u,v)$, and the *diameter* of network $\mathcal{G}$ is defined as the largest eccentricity, *i.e.*, $\mathrm{diam}(\mathcal{G}) := \max_{v\in\mathcal{V}} \epsilon_{\mathcal{G}}(v)$.

As the first step, consider the case that a single copy of antidote is distributed to vertex $c \in \mathcal{V}$ at time 0, *i.e.* $\mathcal{C}_0 = \{c\}$. Then propagation time of the antidote to any specific vertex i can be bounded by the following lemma.

Lemma 2.2. *Let $T_{c,i}$ denote the time that vertex i gets a copy of the antidote originated from vertex $c \in \mathcal{C}_0$. We have*

$$\mathbb{E}(T_{c,i}) \leq \frac{\mathrm{dist}(c,i)}{\gamma}, \tag{2.18}$$

Proof. Let $\{P_k\}_{k=1}^{K}$ denote the set of paths between vertex c and i, such that their lengths are in an ascending order, *i.e.*, $l_k \leq l_{k+1}$. For any $1 \leq k \leq K$, we have

$$dist_{\mathcal{G}}(c,i) \leq l_1 \leq l_k \leq l_K. \tag{2.19}$$

Consequently, the curing time $T_{c,i}$ of vertex i (by vertex c) can be re-written as

$$T_{c,i} = \min_{1\leq k\leq K} T_{c,i}^{k} \leq T_{c,i}^{1}, \tag{2.20}$$

[4] Let $\mathbf{A}$ denote the adjacency matrix of $\mathcal{G}$. The graph Laplacian of $\mathcal{G}$ is defined as $\mathcal{L}(\mathcal{G}) = diag(\vec{D}) - \mathbf{A}$, where $\vec{D}$ is the degree sequence of $\mathcal{G}$, and $diag(\vec{D})$ is the diagonal matrix with $\vec{D}$ as its main diagonal. Since $\mathcal{G}$ is undirected and connected, its Laplacian $\mathcal{L}(\mathcal{G})$ is symmetric and positive-semidefinite, and has n non-negative real eigenvalues. Among these, the second smallest eigenvalue λ_1, referred to as the *algebraic connectivity*, measures the expanding property of $\mathcal{G}$. Particularly, the lower bound of $\eta(\mathcal{G})$ is referred to as the Buser Inequality.

where $T_{c,i}^k$ is the attempted curing time of vertex i, by the antidote copy originated from c, and transmitted along path P_k. For every k, time $T_{c,i}^k$ is the sum of l_k i.i.d. Exponential r.v.'s with mean $\frac{1}{\gamma}$, as a result of which r.v. $T_{c,i}^k$ satisfies Gamma distribution, *i.e.*, $T_{c,i}^k \sim \mathbf{\Gamma}(l_k, \gamma)$, with mean $\mu_k = \frac{l_k}{\gamma}$ and variance $\sigma_k^2 = \frac{l_k}{\gamma^2}$. Then the upper bound in Eq. (2.18) can be obtained through Eq. (2.19) because

$$\mathbb{E}(T_{c,i}) = \mathbb{E}(\min_{1 \le k \le K} T_{c,i}^k) \le \min_{1 \le k \le K} \mathbb{E}(T_{c,i}^k) = \frac{l_1}{\gamma}, \tag{2.21}$$

where $l_1 = \text{dist}_{\mathcal{G}}(c, i)$ is the length of the shortest path between vertex c and i in graph $\mathcal{G}$. □

Lemma 2.2 provides an upper bound of the expected antidote dissemination time from a designated vertex c to vertex i, which is also an upper bound of the curing time of vertex i, if $c \in \mathcal{C}(0)$. Based on this bound, the extinction time of the undesired information can be analyzed with graph augmentation, as illustrated in the following theorem.

Theorem 2.4. *Let $\epsilon_{\mathcal{G}}(v)$ denote the eccentricity of vertex v in graph $\mathcal{G}$. Let $\mathcal{G}_V(\mathcal{V}/V, \mathcal{E}')$ denote the resulting graph induced by contracting vertices in set $V \subset \mathcal{V}$ to a single vertex $f(V)$, and removing all multiple edges*[5]*. Given that antidotes are distributed to the set $\mathcal{C}_0$ at time $t = 0$, The expected extinction time τ_e of the virus can be upper bounded by*

$$\mathbb{E}(\tau_e) \le \frac{1}{\gamma} \begin{cases} \frac{2 \ln K^*_{\mathcal{G}/\mathcal{C}_0}}{1 - (K^*_{\mathcal{G}/\mathcal{C}_0})^{-\frac{1}{\epsilon_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0))}}}, & (w.h.p.) \text{ if } \epsilon_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0)) \le 10, \\ \epsilon_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0)) + \sqrt{(K^s - 1)\epsilon_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0))}, & \text{in general,} \end{cases} \tag{2.22}$$

where $K^*_{\mathcal{G}/\mathcal{C}_0}$ is the number of longest shortest paths starting from the contracted vertex $f(C_0)$ in the quotient graph $\mathcal{G}_V(\mathcal{V}/V, \mathcal{E}')$, and $K^s \ge K^*_{\mathcal{G}/\mathcal{C}_0} + 1$ is the number of shortest paths that start from $f(C_0)$, and are longer than $\epsilon_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0)) - s$.

Remark 2.2. *To prove Theorem 2.4, which upper bounds the extinction time with the curing time through dominant paths, i.e., paths to peripheral vertices, we need to upper/lower bound the maximum/minimum of summations of multiple Exponential variables, i.e., multiple r.v.'s with Gamma distribution, to address which we first present a technical lemma.*

Lemma 2.3. *Let $X_n = \min_{1 \le i \le n} X_i$ be the minimum of n r.v.'s, and $\overline{X_n} = \max_{1 \le i \le n} X_i$ be the maximum of them, where each $X_i \sim \mathbf{\Gamma}(k_i, \theta_i)$, where θ_i's are the rate parameters. Then $\mathbb{E}(X_n)$*

[5] Graph $\mathcal{G}_V$ is the quotient graph of $\mathcal{G}$ through an equivalence relationship induced by partition $\{V, \{v_1\}, \{v_2\}, \cdots\}$ where $v_i \in \mathcal{V} \setminus V$, so that its vertex set $\mathcal{V}/V = (\mathcal{V} \setminus V) \cup f(V)$ has $|\mathcal{V}| - |V| + 1$ vertices.

and $\mathbb{E}(\overline{X_n})$ *are bounded:*

$$\mathbb{E}(\overline{X_n}) \leq \max_i \left\{ \frac{k_i}{\theta_i} \right\} + \left(\frac{n-1}{n} \sum_{i=1}^{n} \frac{k_i}{\theta_i^2} \right)^{1/2}, \tag{2.23}$$

$$\mathbb{E}(X_n) \geq \min_i \left\{ \frac{k_i}{\theta_i} \right\} - \left(\frac{n-1}{n} \sum_{i=1}^{n} \frac{k_i}{\theta_i^2} \right)^{1/2}. \tag{2.24}$$

When $k_i = k$ *and* $\theta_i = \theta$ *for every* $1 \leq i \leq n$ *such that* $\{X_i\}_i$ *are a set of i.i.d Gamma r.v.'s, tighter bounds exist:*

$$\mathbb{E}(\overline{X_n}) \leq \frac{2 \ln n}{\theta(1 - n^{-1/k})}, \tag{2.25}$$

$$\mathbb{E}(X_n) \geq \frac{k \ln 2 - \ln n}{\theta}. \tag{2.26}$$

Proof. Eq. (2.23) and Eq. (2.24) follow from [10, Theorem 2.1] and [10, Corollary 2.1] by substituting the mean $\mu_i = \mathbb{E}(X_i) = \frac{k_i}{\theta_i}$ and variance $\sigma_i^2 = \frac{k_i}{\theta_i^2}$ of Gamma distributions.

For the i.i.d case, Eq. (2.25) is proved in [35, Eq. (7)]. Now we prove the lower bound in Eq. (2.26). Let r.v. $Y_i = -X_i$, then $X_n = \min_i X_i = -\max_i Y_i \triangleq Y_n$. The moment generating function (MGF) of r.v. Y_i can be derived as $M_Y(t) = M_X(-t) = (1 + \frac{t}{\theta})^{-k}$, where $t < \theta$ as required by MGF $M_X(t)$ of r.v. X_i.

Let set $D(M) = \{t \geq 0 | M_Y(t) \geq 1\}$. Then with the technique from [35, Eq. (6)], we have

$$\mathbb{E}(Y_n) \leq \inf_{t \in D(M)} \frac{1}{t} \left[\ln n + \ln M_Y(t) \right] \leq \frac{1}{t} \left[\ln n - k \ln(1 + \frac{t}{\theta}) \right] \triangleq g(t),$$

which is true for every $t \in [0, \theta)$. Notice that $g(t)$ is monotonically decreasing in $[0, \theta]$, so a tighter bound of $\mathbb{E}(Y_n)$ can be upper derived as $g(\theta)$. Then Eq. (2.24) follows from the fact that $E(X_n) = -E(Y_n) \geq -g(\theta)$. □

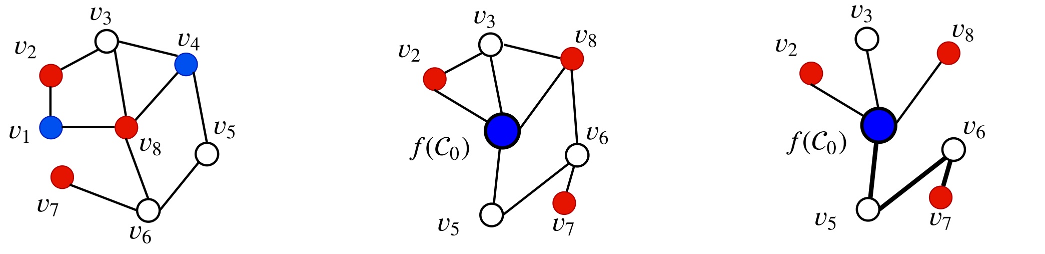
With the preparation of Lemma 2.3, which bounds the maximum and minimum of n Gamma r.v.'s, we are now able to prove the upper bound of extinction time in Theorem 2.4.

Proof. **(Theorem 2.4.)** For an SIC epidemic on graph $\mathcal{G}$, the extinction time of the virus can be bounded by

$$\max_{i \in \mathcal{I}_0} \left\{ \min_{c \in \mathcal{C}_0} T_{c,i} \right\} \leq \tau_e \leq \max_{i \in \mathcal{V} \setminus \mathcal{C}_0} \left\{ \min_{c \in \mathcal{C}_0} T_{c,i} \right\}. \tag{2.27}$$

First we show when $C_0 \geq 2$, the extinction time τ_e on the original graph $\mathcal{G}$ is upper bounded by the extinction time $\widehat{\tau_e}$ of the SIC epidemic on the quotient graph $\mathcal{G}_{\mathcal{C}_0}$, in which one unit of antidote is distributed to vertex $f(\mathcal{C}_0)$ at time $t = 0$.

An example of the graph contraction procedure is shown in Figure 2.8. Consider vertex $v_1, v_4 \in \mathcal{C}_0$. Evolution of the SIC epidemic (spread of virus and antidote) will not be affected by adding an edge (v_1, v_4) to $\mathcal{G}$, no matter such edge (v_1, v_4) exists in $\mathcal{G}$ or not. This is because:



(a) Graph $\mathcal{G}$, $\mathcal{C}_0 = \{v_1, v_4\}$.

(b) Quotient graph $\mathcal{G}_{\mathcal{C}_0}$.

(c) Path to peripheral vertices.

Figure 2.8 Example of contracting cured set $\mathcal{C}_0$ into vertex $f(\mathcal{C}_0)$. In the quotient graph $\mathcal{G}_{\mathcal{C}_0}$, there is $K^*_{\mathcal{G}/\mathcal{C}_0} = 1$ path of length $\epsilon_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0)) = 3$ from $f(\mathcal{C}_0)$ to vertex v_7, and $K' = 1$.

(i) transmission of antidotes between any vertex pair is independent with transmission actions between others; (ii) transmission of antidote between u and v will not result in any state change. Since v_1 and v_4 will stay in *cured* state forever, we can combine and contract them into one vertex $f(\{v_1, v_4\})$ and keep all their edges in $\mathcal{G}$, without affecting the evolution process. The contraction process may result in multiple edges, by removing which the extinction time may increase. Then by induction on the contraction of $\mathcal{C}_0$, the extinction time $\tau_e \le \widehat{\tau_e}$, which is the extinction time of the virus on the quotient graph $\mathcal{G}_{\mathcal{C}_0}$.

Therefore, it is sufficient to consider the SIC epidemic on graph $\mathcal{G}_{\mathcal{C}_0}$, with one unit of antidote distributed to $f(\mathcal{C}_0)$ at time $t = 0$. Then we have the following inequality,

$$\tau_e \le \widehat{\tau_e} := \max_{i \in \mathcal{V} \setminus \mathcal{C}_0} T_{f(\mathcal{C}_0),i} \le \max_{i \in \mathcal{V} \setminus \mathcal{C}_0} T^1_{f(\mathcal{C}_0),i}, \tag{2.28}$$

where $T^1_{f(\mathcal{C}_0),i} \sim \mathbf{\Gamma}\left(\mathrm{dist}_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0), i), \gamma\right)$. The last inequality of Eq. (2.28) follows from Eq. (2.20) in Lemma 2.2. Next we discuss $\max_{i \in \mathcal{V} \setminus \mathcal{C}_0} T^1_{f(\mathcal{C}_0),i}$ and the eccentricity $\epsilon_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0))$ of the central cured hub $f(\mathcal{C}_0)$.

By definition, distance between any vertex i and the cured hub $f(\mathcal{C}_0)$ satisfies

$$\mathrm{dist}_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0), i) \le \epsilon_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0)) \le \mathrm{diam}(\mathcal{G}_{\mathcal{C}_0}) \le \mathrm{diam}(\mathcal{G}). \tag{2.29}$$

Without loss of generality, suppose $\epsilon_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0)) = k + s$ is achieved by the path from $f(\mathcal{C}_0)$ to vertex u in the quotient graph $\mathcal{G}_{\mathcal{C}_0}$. Consider infected vertex i that is $\mathrm{dist}_{\mathcal{G}_{\mathcal{C}_0}}(i, f(\mathcal{C}_0)) = k$ hops away from the cured hub $f(\mathcal{C}_0)$, such that $s \ge 1$. Let r.v. X and Y denote the time it takes a copy of antidote to reach vertex u and v from the hub $f(\mathcal{C}_0)$, respectively. We first show that when $k + s$ is small, the probability that $X \le Y$ is small.

Clearly, $X \sim \mathbf{\Gamma}(k+s, \theta)$ and $Y \sim \mathbf{\Gamma}(k, \theta)$, so $\frac{X}{X+Y}$ satisfies Beta distribution with parameters $k + s$ and k, and

$$\mathbb{P}(X \le Y) = \mathbb{P}(\frac{X}{X+Y} \le 0.5) = \frac{\Gamma(2k+s)}{\Gamma(k+s)\Gamma(k)} \int_0^{0.5} t^{k+s-1}(1-t)^{k-1} dt, \tag{2.30}$$

where $\Gamma(k) = \int_0^\infty x^{k-1} e^{-x} dx$ is the Gamma function. It is difficult to bound Eq. (2.30) due to the integral, so we plot this probability for k and s values, such that $k + s$ is in the range of frequently[6] seen network diameters, as shown in the following Figure 2.9.

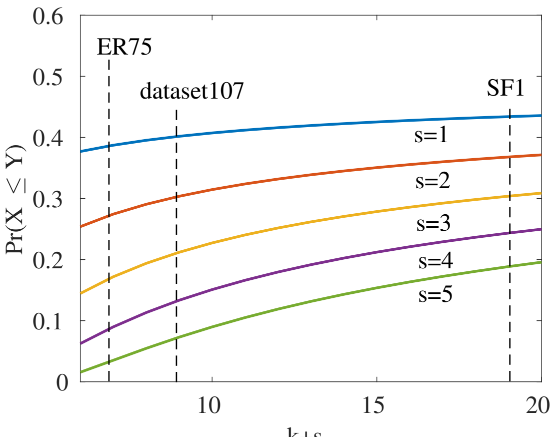


Figure 2.9 Probability $\mathbb{P}(X \leq Y)$ for $k + s$ ($\leq$ graph diameter diam($\mathcal{G}$)) ranging from 5 to 20.

In Figure 2.9, the dashed vertical lines identify the diameter diam($\mathcal{G}$) of three networks used for simulation validation in this section. Among these `dataset107` is a real-world network portion, whose diameter is typical. When $k+s \leq \text{diam}(\mathcal{G}) \leq 10$, the probability that the antidote will reach u sooner than i, who is s-hops closer than u from $f(\mathcal{C}_0)$ is small (< 0.2) when $s \geq 3$. In addition, the rate parameter θ of the Gamma distribution equals to the curing rate γ, so when γ is also small, the gap between the two time intervals Y and X is therefore small as well, such that we only need to consider vertices who are located the furthest from vertex $f(\mathcal{C}_0)$ in graph $\mathcal{G}_{\mathcal{C}_0}$, *i.e.*, the $K^*_{\mathcal{G}/\mathcal{C}_0}$ distinct vertices, such as u, satisfying $dist_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0), i) = \epsilon_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0))$. Then the first term on the right-hand side of Eq. (2.22) follows from the tighter upper bound Eq. (2.25) in Lemma 2.3. Note that when $K^*_{\mathcal{G}/\mathcal{C}_0} = 1$, the first term is not well-defined, because the denominator equals to 0. In this case, we take $\mathbb{E}(\tau_e) \leq \frac{\epsilon_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0))}{\gamma}$ instead.

If this condition is not satisfied, *e.g.*, when diam($\mathcal{G}$) is large, such that $\mathbb{P}(X \leq Y)$ can not be omitted, we always have the option to consider more paths, which are possibly shorter, to upper bound the extinction time. Then the upper bound (second term in Eq. (2.22)) follows from the upper bound of general distributions, which is illustrated in Eq. (2.23) of Lemma 2.3. □

Note that the eccentricity $\epsilon_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0))$ is a property of the augmented quotient graph $\mathcal{G}_{\mathcal{C}_0}$. To apply Theorem 2.4, we need to know the exact locations (vertices) where copies of antidotes are disseminated, *i.e.*, set $\mathcal{C}_0$. If such knowledge is not readily available, the extinction time can still be bounded by the diameter (or more generally, distribution of vertex eccentricities)

[6] In most networks we consider (complete graph, star graph, ER graph and scale-free network etc.), the diameter diam($\mathcal{G}$) $= O(\log |\mathcal{V}|)$ is small due to the *small-world* effect, so probability $\mathbb{P}(X \leq Y)$ in Eq. (2.30) is also small.

of graph $\mathcal{G}$, as given in the following two corollaries. Proof of Corollary 2.2 and 2.4 are simple as they directly follow from the fact that $\epsilon_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0)) \leq \min_{c \in \mathcal{C}_0} \epsilon_{\mathcal{G}}(c) \leq \text{diam}(\mathcal{G})$.

Corollary 2.2. *Particularly when $C_0 = 1$, for any initial antidote recipient $c \in \mathcal{V}$,*

$$\mathbb{E}(\tau_e) \leq \frac{1}{\gamma}\left[\text{diam}(\mathcal{G}) + \sqrt{\text{diam}(\mathcal{G})(|\text{Peri}(\mathcal{G})|/2 - 1)}\right], \tag{2.31}$$

where $\text{Peri}(\mathcal{G}) = \{v \in \mathcal{V} \mid \epsilon(v) = \text{diam}(\mathcal{G})\}$ *is the set of peripheral vetices in $\mathcal{G}$.*

Corollary 2.2 is also an upper bound of the expected extinction time $\mathbb{E}(\tau_e)$ for the case of $C_0 > 1$, because the more copies of antidote distributed initially at $t = 0$, the less (statistically bounded) time it takes to fully remove virus from the network, *i.e.*, a shorter extinction time. A direct implication of Corollary 2.2 is that for small-world graphs, which naturally emerges in various contexts, such as social networks, the extinction time is $O(\frac{\log n}{\gamma})$, due to the $O(\log n)$ scaling of the network diameter. Though not a tight bound compared to Theorem 2.3, this corollary is much more accessible in the sense of computation complexity, especially for large networks. Obtaining the eccentricity distribution of a graph, including diameter and peripheral size, is at most $O(n|\mathcal{E}|)$ in time complexity with further speedups [112], which is much faster than obtaining the Cheeger constant $\eta(\mathcal{G})$. By the eccentricity distribution and the initial cured count C_0, a more accurate upper bound can be derived as follows.

Corollary 2.3. *Suppose vertices in $\mathcal{C}_0$ are chosen uniformly at random from $\mathcal{V}$, and cumulative (probability) density function (CDF) of the eccentricity distribution of graph $\mathcal{G}$ is $F(x)$ ($\text{radi}(\mathcal{G}) \leq x \leq \text{diam}(\mathcal{G})$). The expected extinction time can be upper bounded by*

$$\mathbb{E}(\tau_e) \leq \frac{1}{\gamma}\left[\mu_{C_0} + \sqrt{\mu_{C_0}\left(\frac{n - C_0}{\text{diam}(\mathcal{G})} - 1\right)}\right], \tag{2.32}$$

where $\mu_{C_0} = \sum_{k=\text{radi}(\mathcal{G})}^{\text{diam}(\mathcal{G})} k[1 - F(k)]^{C_0}$ is the expected maximum eccentricity of vertices in the cured set $\mathcal{C}_0$, and radius $\text{radi}(\mathcal{G})$ *equals to the minimum eccentricity in $\mathcal{G}$.*

Corollary 2.3 stems from Theorem 2.4, by considering the expected maximum eccentricity of the C_0 antidote recipients in $\mathcal{C}_0$, given that antidotes are randomly distributed at time 0. Specifically, for any vertex $v \in V \setminus \mathcal{C}_0$, the distance between i and vertex $f(\mathcal{C}_0)$ in the quotient graph $\mathcal{G}_{\mathcal{C}_0}$ satisfies $dist_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0), v) \leq \epsilon_{\mathcal{G}_{\mathcal{C}_0}}(f(\mathcal{C}_0)) \leq \min_{c \in \mathcal{C}_0} \epsilon_{\mathcal{G}}(c) \leq \text{diam}(\mathcal{G})$, which is the diameter of the original graph $\mathcal{G}$. On the other hand, changing perspective to the initial infected set $\mathcal{I}_0$, in which vertices are in infected state at time 0, the expected extinction time $\mathbb{E}(\tau_e)$ can also be bounded below by considering the shortest paths between vertices in set $\mathcal{I}_0$ and set $\mathcal{C}_0$.

Corollary 2.4. *Given the initial infected set $\mathcal{I}_0 = \mathcal{I}(t = 0)$, the expected extinction time can be lower bounded by*

$$\mathbb{E}(\tau_e) \geq \frac{1}{\gamma}\left[\text{dist}(\mathcal{C}_0, i^*) \ln 2 - lnK_{i^*}\right], \tag{2.33}$$

where $\mathrm{dist}(\mathcal{C}_0, i) = \min_{c\in\mathcal{C}_0}\{\mathrm{dist}(c,i)\}$ *denote the shortest distance between set* $\mathcal{C}_0$ *and vertex* i*,* $i^* = \arg\max_{i\in\mathcal{I}_0}\{\mathrm{dist}(\mathcal{C}_0, i)\}$*, and* K_{i^*} *is the number of shortest paths*[7] *between set* $\mathcal{C}_0$ *and* i_**.*

Proof. From Eq. 2.27, we have

$$\tau_e \geq \max_{i\in\mathcal{I}_0}\left\{\min_{c\in\mathcal{C}_0} T_{c,i}\right\} \geq \min_{c\in\mathcal{C}_0} T_{c,i^*}, \tag{2.34}$$

where $T_{c,i^*} = \min_{1\leq k\leq K_{i^*}} T^k_{c,i^*}$ is the minimum of K_{i^*} i.i.d. Gamma r.v.'s, and each $T^k_{c,i^*} \sim \mathbf{\Gamma}(\mathrm{dist}(\mathcal{C}_0, i), \frac{1}{\gamma})$, which comes from Lemma 2.2. Then the lower bound in Eq. (2.33) follows immediately from Eq. (2.26) in Lemma 2.3. □

2.4.3 Validation in Synthetic and Real-world Networks

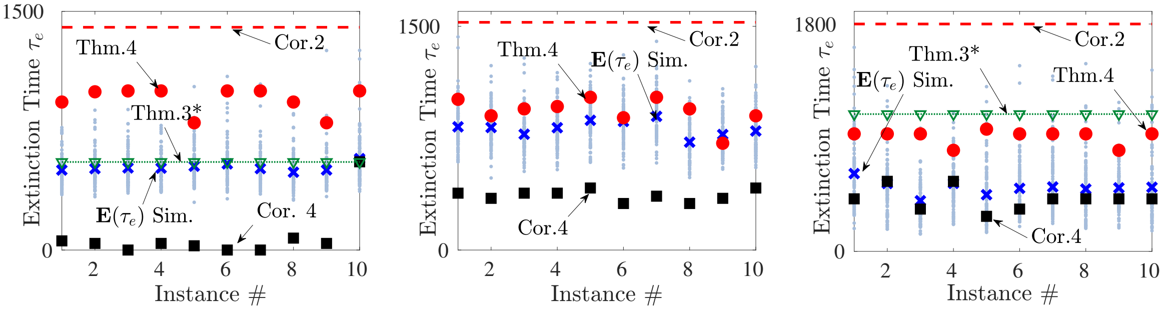
To validate the derived bounds, especially the extinction time of the undesired information, we test three networks with different topologies, but similar sizes: random graph (ER graph ER75 with edge connecting probability $p = 0.0075$, scale-free graph SF1 generated with the Barabási–Albert model by attaching one edge for one vertex each time, and a fraction of the real-world Facebook friendship network `dataset107` [64]. Statistics of the three connected graphs, ER75, SF1 and `dataset107` are shown in Table 2.1. Though all of the three networks have around 1000 vertices, the SF1 network has a large diameter ($\mathrm{diam}(\mathcal{G}) > 10$, which indicates that the condition in the first case of Theorem 2.4 does not hold.) and is less connected, compared to ER75 and `dataset107`. This can be observed by the low average degree and algebraic connectivity λ_1. ER75 is more clustered and even, in the sense that it achieves a high algebraic connectivity λ_1 with a relatively low average degree.

Table 2.1 Statistics of networks used in simulation (Section 2.4.3).

Property	Description	ER75	SF1	`dataset107`
n	network size $\lvert\mathcal{V}\rvert$	1000	1000	1033
$\bar{d}$	average degree	7.362	1.9980	51.7851
$\mathrm{diam}(\mathcal{G})$	network diameter, $\max_{v\in\mathcal{V}} \epsilon_{\mathcal{G}}(v)$	7	19	9
$\mathrm{radi}(\mathcal{G})$	network radius, $\min_{v\in\mathcal{V}} \epsilon_{\mathcal{G}}(v)$	5	10	5
λ_1	algebraic connectivity	0.7618	0.002	0.1329
$\lvert\mathrm{Peri}(\mathcal{G})\rvert$	periphery size, $\lvert\{v \in \mathcal{V} \mid \epsilon_{\mathcal{G}}(v) = \mathrm{diam}(\mathcal{G})\}\rvert$	1	8	1

Figure 2.10a-c show the simulated extinction time (grey dots) in the three networks, respectively, whose mean $\mathbb{E}(\tau_e)$ are identified with blue '×' markers. Each instance in the x-axis corresponds to the same initial cured set $\mathcal{C}_0$ and infected set $\mathcal{I}_0$, and each instance is repeated

[7] In practice, obtaining quantity K_{i^*} requires executing path searches repetitively. To avoid high computation load in the simulation, we use the upper bound of K_{i^*}, that is, $\lvert\{i \in \mathcal{I}_0 \mid \mathrm{dist}(\mathcal{C}_0, i) = \mathrm{dist}(\mathcal{C}_0, i^*)\}\rvert$ when evaluating the lower bound in Corollary 2.4.



(a) ER75 $C_0 = 3$, $\gamma = 0.005$. **(b)** SF1 $C_0 = 3$, $\gamma = 0.02$. **(c)** `dataset107` $C_0 = 5$, $\gamma = 0.005$.

Figure 2.10 Bounds and simulation of the extinction time $\mathbb{E}(\tau_e)$ in general networks. The curing rate γ in the SF1 scenario is increased to avoid a lenthy simulation in SF1.

100 times to obtain $\mathbb{E}(\tau_e)$. The red round markers and black squares correspond to the upper bounds in Theorem 2.4 and Corollary 2.4, respectively, both of which rely on vertex eccentricity properties of the network. The red dashed line and green dashed line with triangle markers correspond to the upper bounds presented in Corollary 2.2 that relies on network diameter diam($\mathcal{G}$), and Theorem 2.3 that relies on the Cheeger constant[8] $\eta(\mathcal{G})$, respectively.

We highlight the following observations:

(1) *Network Topology*: Extinction time $\mathbb{E}(\tau_e)$ varies more violently in SF1 and `dataset107` networks that are less 'regular' (less like clique topology) in the sense that vertices differ more in degree, centrality, etc. (indicating different importance/influential in status of individuals). Therefore, in SF1, due to the larger curing rate γ, and the larger diameter diam($\mathcal{G}$) ,the probability in Eq. (2.30) will not be small enough to be neglected, so the upper bound specifically for i.i.d. Gamma distributions (first term in the right-hand side of Eq. (2.22)) in Theorem 2.4 does not hold any more, but the second term for general distributions still holds, which is adopted as the red round markers in Figure 2.10b. For the 'regular' ER75 and `dataset107` networks, despite their differences in average degree, and algebraic connectivity, their similarity in the vertex eccentricity properties (diameter, radius, and periphery size in Table 2.1) leads to very close extinction time, which indicates the usefulness of Corollary 2.2.

(2) *Tightness of the Bounds*: Theorem 2.4 and Corollary 2.4 are tighter in the 'irregular' SF1 and `dataset107`, due to the existence of few 'dominant' longest shortest paths, *i.e.*, K^* is small. In contrast, for the ER75 newtork, vertex degrees and pair-wise distances are highly homogeneous (small variance in degree distribution), resulting in a large K^* that affects the tightness of both upper and lower bounds. Corollary 2.2 (red dashed line in Figure 2.10a-c) that relates to the network diameter diam($\mathcal{G}$) is an upper bound not only good for the mean,

[8] Approximating the Cheeger constant $\eta(\mathcal{G})$ with the lower bound $\frac{\lambda_1}{2}$ results in a very loose bound of $\mathbb{E}(\tau_e)$ at $\sim 10^4$, so it is not shown for a better illustration and comparison among other bounds. The green dashed line corresponds to the upper bound in Cheeger's Inequality $\eta(\mathcal{G}) = 2\sqrt{\lambda_1}$, which is presented to illustrate how close Theorem 2.3 can be, if the Cheeger constant reaches its maximum possible value. Hence the star sign in the legend.

but also for every simulation runs. Theorem 2.3 is not a tight bound. As indicated by the green dashed line with triangle markers, which approximates the Cheeger constant with its maximum possible value $\eta(\mathcal{G}) = 2\sqrt{\lambda_1}$, even so the derived upper bound is not as close as Theorem 2.4, especially in SF1 and `dataset107`.

(3) *Implications*: From the proof of Theorem 2.4, another implication is that, the extinction time of the undesired information can be bounded by the number and the length of shortest paths, which go through vertices of the initial cured set $\mathcal{C}_0$. This observation sheds light on antidote distribution (injection of desired information), when used as a counter-measure against the epidemic spreading of undesired information: A general guideline on selecting vertices for set $\mathcal{C}_0$ is to choose vertices that sit on the most number of shortest paths, *i.e.*, vertices with high *betweenness centrality*. To better understand the structural change due to $\mathcal{C}_0$ and its impact to the extinction time of the virus, we discuss antidote distribution in the next section in details.

2.5 Divide-and-Conquer: Leveraging Topology to Control Undesired Information

As discussed in the introduction, desired information, *e.g.*, security patches, can be purposely injected into networked systems to *control* the epidemic spreading of undesired information, *e.g.*, computer malware. There has been extensive studies on control of epidemic information propagation, most of which model the control process as an optimization problem under resource constraints. Preciado et.al. developed a convex framework[96] to evaluate the optimal allocation of edge control, immunization and non-replicable antidotes resources, in which the network is modeled as a directed graph. Borgs *et.al.* studied the optimal distribution of non-replicable antidote[19] given that the curing rate, proportional to the units of given antidote, is non-uniform. As for replicable antidote, Khouzani *et.al.* formulated the control strategy[57] with both replicable and non-replicable antidote into an optimal resource allocation problem. Chen utilized optimal control theory to determine the optimal distribution time[25] of a replicable antidote for timely control. However in [19, 57, 25], the information dynamic in the system is described by nonlinear differential equations, which indicates it is a *population dynamic*, rather than a *network dynamic*, or equivalently, topology of the network is not taken into consideration. On the other hand, the influence of network topology is studied in terms of epidemic threshold[61] under immunization and spreading/extinction time[59] under SI or SIS epidemic propagation models, while the dissemination of replicable antidote is not incorporated. Therefore, in this section we explore the *distribution strategies of replicable antidote*, leveraging network *topology* in order to control the undesired information.

Apparently, both the extinction time and the half-life time changes with the initial distribution choice, *i.e.*, $\mathcal{C}(0)$, so a carefully chosen set of vertices can effectively shorten the lifetime of the virus. This is shown by a simple example in Figure 2.11. As can be seen, after the initial distribution, the potential hazard zone (pink shaded region) in the random distribution case is

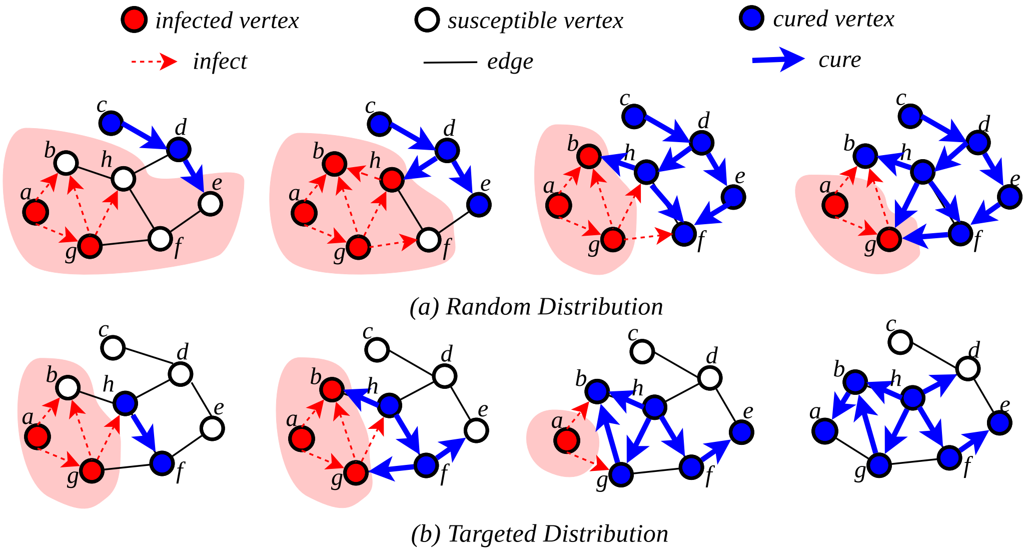


Figure 2.11 An example of (a) random distribution; (b) targeted distribution.

larger than that of the targeted distribution. During the propagation process, the effect of the replicable antidote is two-fold: on individual bases, it cures infected vertices which decreases the infection count; on the other hand, the expanding cured set composes a structure to retain the potential hazard zone of the virus. In the latter case the influence of network topology is more evident because the dynamics is changing a topological property of the system.

2.5.1 Topology-based Antidote Distribution

As shown in the previous example of Figure 2.11b, the potential hazard zone of the virus is restricted to a limited region of the network, *i.e.* a non-empty set of susceptible vertices are 'quarantined' by the initial antidote distribution, who will never be infected during the dynamics. We are interested in such *locking* condition, which effectively restrains the virus propagation. To characterize this effect, we introduce the *initial locking time*, defined as follows.

Definition 2.3. *Let $\mathcal{G}^*(t)$ be the induced subgraph of $\mathcal{G}$ by removing the cured vertices $\mathcal{C}(t)$. We write $\mathcal{G}^*(t) = \cup_{1\leq i\leq k(t)} G_i$, where $G_i(V_i, E_i)$ are components of $\mathcal{G}^*(t)$ and $k(t)$ denote the number of components at time t. The* ***initial locking time*** *τ_0 is defined as the first time that $\mathcal{G}^*(t)$ becomes disconnected, or equivalently*

$$\tau_0 = \sup\{t > 0 \mid k(t) \geq 2\}.$$

Remark 2.3. *Time instant τ_0 marks a critical point of the virus epidemic, because a topological property of the remaining graph $\mathcal{G}^*(t)$, i.e., the connectivity of $\mathcal{G}^*(t)$, changes at τ_0.*

When $t < \tau_0$, $k(t) = 1$, the virus can potentially spread to every corner of the remaining graph $\mathcal{G}^*(t)$. As time t goes beyond τ_0, further fragmentation starts to happen in each $G_i \subset$

$\mathcal{G}^*(t)$, and potential hazard of the virus can be treated as under control. Then we have the following theorem regarding the extinction time $\mathbb{E}(\tau_e)$ and $\mathbb{E}(\tau_0)$.

Theorem 2.5. *Let $C_0 = |\mathcal{C}(0)|$,* diam(G) *and $\eta(G)$ denote the diameter and Cheeger constant of graph G, respectively. Then the expected extinction time can be upper bounded by*

$$\mathbb{E}(\tau_e) \leq \mathbb{E}(\tau_0) + \frac{1}{\gamma}[\frac{2}{(n-C_0)\eta(\mathcal{G})} + \max_{1\leq i\leq k(\tau_0)}\{\mathrm{diam}(G_i)\}]. \tag{2.35}$$

Proof. Consider r.v. Z_i and Y_i, defined as follows.

$$Z_i := \inf_{t>\tau_0}\left\{t-\tau_0 \mid |V_i \cap \mathcal{I}(t)| = 1\right\}, \tag{2.36}$$

$$Y_i := \inf_{t>Z_i+\tau_0}\{t-(Z_i+\tau_0) \mid |V_i \cap \mathcal{I}(t)| = 0\}. \tag{2.37}$$

Now we re-write $\mathbb{E}(\tau_e)$ with $\mathbb{E}(\tau_0)$,

$$\begin{aligned}\mathbb{E}(\tau_e) \quad &\leq \quad \mathbb{E}(\tau_0) + \mathbb{E}(\max_{1\leq i\leq k(\tau_0)}\{Z_i+Y_i\}) \\ &\overset{Jensen}{\leq} \mathbb{E}(\tau_0) + \max_{1\leq i\leq k(\tau_0)}\{\mathbb{E}(Z_i)+\mathbb{E}(Y_i)\} \\ &\leq \mathbb{E}(\tau_0) + \max_{1\leq i\leq k(\tau_0)}\{\mathbb{E}(Z_i)\} + \max_{1\leq i\leq k(\tau_0)}\{\mathbb{E}(Y_i)\}),\end{aligned} \tag{2.38}$$

Note that the first inequality of Eq. (2.38) follows from the fact that among all the components of $\mathcal{G}^*(\tau_0)$, some may not contain infected vertices. The physical meaning of Z_i is the time interval between the initial locking and the first vertex in G_i is cured, or equivalently, the minimum of $\delta(G_i)$ i.i.d exponential r.v.'s with parameter γ, where $\delta(G_i)$ denote the number of edges in edge cut $[V_i, \mathcal{V}\setminus V_i]$. Hence $Z_i \sim Exp(\delta(G_i)\gamma)$. Therefore

$$\begin{aligned}\max_{1\leq i\leq k(\tau_0)}\{\mathbb{E}(Z_i)\}) &= \frac{1}{\gamma}[\max_{1\leq i\leq k(\tau_0)}\{\frac{1}{|\delta(G_i)|}\} \\ &\leq \frac{1}{\gamma}[\max_{1\leq i\leq k(\tau_0)}\{\frac{1}{\eta(\mathcal{G})\min\{|V_i|, n-|V_i|\}}\} \\ &\leq \frac{1}{\gamma}\frac{2}{(n-C_0)\eta(\mathcal{G})}.\end{aligned} \tag{2.39}$$

For Y_i, it denotes the time interval between τ_0+Z_i and the time that all the infected vertices in G_i are cured. Suppose the first cured vertex in G_i is $v_i \in V_i$, then Y_i is bounded above by the spreading time of the antidote along the shortest-path spanning tree of G_i rooted at v_i. Since $v_i \in {}_i$ can be any vertex that is directed connected to $\mathcal{C}(\tau_0)$. Then

$$Y_i \leq \max_{v_i\in E_i}\{\sum_{s=1}^{depth(G_i,v_i)} \Delta_C^s\} \leq \sum_{s=1}^{\mathrm{diam}(G_i)} \Delta_C^s, \tag{2.40}$$

where $\Delta_C^s \sim Exp(\gamma)$ denote the time intervals between the $s-1$-th and the s-th curing,

$depth(G_i, v_i)$ denotes the depth of the shortest-path spanning tree of G_i, and $\mathrm{diam}(G_i)$ is the diameter (length of the longest shortest-path) of G_i. The second inequality of Eq. (2.40) follows from the fact that $depth(G_i, v_i) \leq \mathrm{diam}(G_i)$, $\forall v_i \in E_i$, so $\sum_{s=1}^{depth(G_i,v_i)} \Delta_C^s \leq \sum_{s=1}^{\mathrm{diam}(G_i)} \Delta_C^s$, for any $v_i \in E_i$. Therefore,

$$\max_{1\leq i\leq k(\tau_0)}\{\mathbb{E}(Y_i)\}) \leq \frac{1}{\gamma}[\max_{1\leq i\leq k(\tau_0)}\{\mathrm{diam}(G_i)\}] \tag{2.41}$$

Combining Eq. (2.38), (2.39) and (2.41) completes the proof. □

Theorem 2.5 implies that, the virus infections are defeated through a *divide-and-conquer* procedure. Specifically, the extinction time can be mainly determined by quantity $\mathbb{E}(\tau_0)$ and quantity $max_{1\leq i\leq k(\tau_0)}\{\mathrm{diam}(G_i)\}$, since $\eta(G) \geq \frac{n}{2}$ (the case that all vertices form a single line) indicates the second term in Eq. (2.35) is at most $O(1)$.

2.5.2 Ideal Antidote Distribution Policy

From a holistic view, there are two influential factors on the extinction time: the initial antidote distribution policy, and the network topology. The former includes the initial cured count C_0 and the recipient of the C_0 antidotes, *i.e.* assignment of $\mathcal{C}(0)$. Now consider the initial locking time τ_0. Apparently it is decreasing as the initial cured count C_0 increases, as well as $\mathrm{diam}(G_i)$. However, it is not reasonable nor realistic to let C_0 approach n. We want to find the most effective way to distribute as less antidote as possible, under condition that the extinction time can be mostly shortened. So in this subsection, we assume that we have just enough antidote such that $\tau_0 = 0$, *i.e.* $\mathcal{C}(0)$ is a *vertex cut* of $\mathcal{G}$. Then we define the following metric to describe the impact of the graph topology on the extinction time with respect to an assignment of $\mathcal{C}(0)$.

Definition 2.4. *The hazard index $\phi(\mathcal{C}(0))$ is defined as the maximum diameter of components of the initially disconnected graph $\mathcal{G}^*(t) = \cup_{1\leq i\leq k(t)} G_i$,*

$$\phi(\mathcal{C}(0)) = \max\{\mathrm{diam}(G_i) \mid G_i \subset \mathcal{G}^*(\tau_0), 1 \leq i \leq k(\tau_0)\}.$$

Further, we provide the following upper bound of $\phi(\mathcal{C}(0))$ in terms of $|V_i|$, the number of vertices in each components of $\mathcal{G}^*(0)$.

Theorem 2.6. *Let $d'(\mathcal{G})$ denote the minimum degree of graph $\mathcal{G}$, then*

$$\phi(\mathcal{C}(0)) \leq \frac{3}{d'(\mathcal{G})+1}\max_{1\leq i\leq k(\tau_0)}\{|V_i|\} - 1. \tag{2.42}$$

Proof. Let the minimum degree of each component G_i be denoted as d_i'. If it satisfies $d_i' \leq \frac{|V_i|-2}{2}$

($d'_i > \frac{|V_i|-2}{2}$ is highly unlikely, because it would suggest every component G_i is dense), then[21]

$$\mathrm{diam}(G_i) \leq 3\lfloor \frac{|V_i|}{d'_i+1} \rfloor - \begin{cases} 3, & n \mod (d'_i+1) = 0; \\ 2, & n \mod (d'_i+1) = 1; \\ 1, & \text{otherwise.} \end{cases} \tag{2.43}$$

Then

$$\begin{aligned} \phi(\mathcal{C}(0)) &\leq \max_{1 \leq i \leq k(\tau_0)} \{3\lfloor \frac{|V_i|}{d'_i+1} \rfloor - 1\} \leq \max_{1 \leq i \leq k(\tau_0)} \{3(\frac{|V_i|}{d'(\mathcal{G}^*(\tau_0))+1}) - 1\} \\ &\overset{\text{Note}}{\leq} \max_{1 \leq i \leq k(\tau_0)} \{3(\frac{|V_i|}{d'(\mathcal{G})+1}) - 1\} \leq \frac{3}{d'(\mathcal{G})+1} \max_{1 \leq i \leq k(\tau_0)} \{|V_i|\} - 1. \end{aligned} \tag{2.44}$$

Note that the third inequality in Eq. (2.44) is not strict, but still reasonable. Since $d'(\mathcal{G}^*(\tau_0)) \geq d'(\mathcal{G})$ holds except the case that the vertext with the minimal degree (denote as h) is directly adjacent to a vertex in $\mathcal{C}(0)$. If the degree of h is decreased much due to the removal of $\mathcal{C}(0)$, then it would be more convenient to include h in $\mathcal{C}(0)$ at the initial antidote distribution, and in this case, the degree of the remaining vertices is decreased at most 1, due to the fact that $\mathcal{G}$ is simple, that is, $\mathcal{G}$ does not have repetitive edges. □

Theorem 2.6 provides an upper bound of $\phi(\mathcal{C}(0))$ in terms of the size of the giant component $\max_{1 \leq i \leq k(\tau_0)}(|V_i|)$ in $\mathcal{G}(\tau_0)$, where $\tau_0 = 0$ in this case. In discrete mathematics, this is equivalent to the *most balanced cut* problem, that is, finding such vertex-cut constrained on different balance requirements, which in our case, is the number of vertices in each component G_i.

On the other hand, clearly from Eq. (2.35), our goal of assigning $\mathcal{C}(0)$ is to minimize the hazard index, *i.e.* $\min_{\mathcal{C}(0)}\{\phi(\mathcal{C}(0))\}$. From the Moore Bound [81] Inequality, we know that the diameter of each subgraph G_i can be lower bounded by[9]

$$\mathrm{diam}(G_i) \geq \begin{cases} \frac{|V_i|-2}{2}, & d_i = 2, \\ \log_{d_i-1}[(|V_i|-1)\frac{d_i-2}{d_i}) + 1], & d_i > 2, \end{cases}$$

where d_i is the maximum degree and $|V_i|$ is the number of vertices of graph G_i respectively. Let $f(d_i, |V_i|) := \log_{d_i-1}[(|V_i|-1)\frac{d_i-2}{d_i}) + 1]$. We can show that $f(d_i, |V_i|)$ is decreasing in d_i, but increasing in $|V_i|$. In the mean time, $\frac{|V_i|-2}{2}$ is also increasing in $|V_i|$, irrelevant to d_i. Consequently,

$$f(d_i, |V_i|) \geq f(d(\mathcal{G}), |V_i|),$$

where $d(\mathcal{G})$ denotes the maximum degree of the network $\mathcal{G}$.

Now minimizing $\max\{\mathrm{diam}(G_i)\}$ becomes minimizing $\max\{|V_i|\}$, that is, we want to find a minimum vertex cut $\mathcal{C}(0) \subset \mathcal{V}$, such that in the induced subgraph $\mathcal{G}^*(\tau_0) = \mathcal{G} \setminus \mathcal{C}(0)$,

[9] We remark that the Moore Bound is fairly difficult to attain, but it is a valid lower bound with respect to the number of vertices of each component.

$\min_{\mathcal{C}(0)}\{\max_i\{|V_i|\}\}$ can be achieved. Again, the upper bound of $\phi(\mathcal{C}(0))$, (and hence that of $\mathbb{E}(\tau_e)$) is related to the maximum number of vertices in each component of $\mathcal{G}^*(\tau_0)$, which also leads to the *minimum most balanced vertex-cut* problem. By assigning $\mathcal{C}(0)$ to the minimum most balanced cut, the upper bound of $\mathbb{E}(\tau_e)$ in Eq. (2.35) can be tightened because $\max_i\{\text{diam}(G_i)\}$ is tightened.

Therefore, to maximize the effect of antidote that can propagate and spread itself, as well as to minimize the extinction time, the target of an ideal initial antidote distribution strategy is to assign $\mathcal{C}(0)$ to the minimum most balanced vertext-cut of the network $\mathcal{G}$.

2.5.3 Practical Approaches

In real-world networks that are complex in topology, however, it is not possible to find a small vertex-cut, let alone a minimum most balanced vertex-cut. For example, in a complete graph K_n, the minimum vertex-cut contains $n-1$ vertices. It translates to either a long τ_0, or a large enough C_0, but $\mathbb{E}(Y_i) \simeq \frac{1}{\gamma(n-1)}$, in which case the upper bounds in the previous sections are more applicable. In addition, searching for the minimum most balanced cut or directly searching for vertices whose removal will result in a smaller $\phi(\mathcal{C}(0))$ is difficult. In fact, finding the minimum most balanced cut for general graphs is NP-Hard [58]. So we introduce the following applicable and practical approaches under the guideline of minimizing $\phi(\mathcal{C}(0))$, or minimizing the size of the giant component in $\mathcal{G}^*(\tau_0)$.

2.5.3.1 betcen-based approach

Betweenness centrality of a vertex v is defined as $g(v) = \sum_{s\neq v\neq t} \frac{\delta_{st}(v)}{\delta_{st}}$, where δ_{st} denotes the number of shortest paths between vertex s and v, while $\delta_{st}(v)$ are the number of those paths that passes through vertex v. Parameter $g(v)$ indicates how likely vertex v sits in other vertices' shortest paths. Removal of a vertex with high g value will likely break more shortest paths, which is an effective measure as indicated by the extinction time bound in the previous section. However, it requires global knowledge and takes $O(|\mathcal{V}||\mathcal{E}|)$ [20] time to calculate.

2.5.3.2 ccfs-based approach

Clustering Coefficient of a vertex v is defined as $C(v) = \frac{2|\{e_{st}|e_{st},e_{vs},e_{vt}\in\mathbb{E}\}|}{d(v)[d(v)-1]}$, which shows how densely connected is v's neighborhood. To calculate $C(v)$, knowledge of vertices within the distance of two to v is required. The reason of using C value as an indicator is that when $C(v)$ is a small but strictly positive value, it implies that v's neighbor is not well-connected, and relies v to function as a bridge between its neighbors. This is especially true when the graph does not have many 'long edges'.

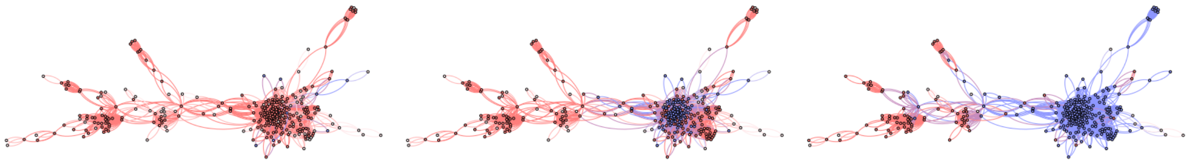


Figure 2.12 A example of an SIC dynamics ($\beta = \gamma = 0.003$, $I_0 = 200$) after the initial distribution of $C_0 = 40$ copies of antidotes with ccfs-based approach.

2.5.3.3 degree-based approach

Degree of vertex v, $d(v)$ is easy to attain because it only requires knowledge of one-hop neighbors of a vertex. Higher degree indicates a vertex has a higher chance of being a hub. Hence the removal of such a vertex will result in the removal of a lot of edges.

In the implementation of these approaches in the next subsection, we first calculate the g, C and d values for each vertex, and then sort them to find the best candidates. Considering the sorting process require global knowledge of the graph, in real world implementation, the sorting process can be substituted with a cut-off mechanism with predetermined threshold values.

2.5.4 Numerical Results and Discussion

To validate and compare the proposed approaches, we first analyze an extreme case scenario, and then present simulation results in network portions acquired from the real world OSN, Facebook. Figure 2.12 shows a set of snapshots of an SIC information dynamics evolution in dataset0, where color red, white and blue indicates infected, susceptible and cured respectively.

Consider first a special case when $\mathcal{G} = S_n$, i.e. the star network with one hub and $n-1$ peripherals. Based on all three approaches, the hub will be the first one selected in $\mathcal{C}(0)$, due to its high g and d value, as well as its low C value. When the hub is cured, the SIC dynamic is in a *locked* condition, that is, the SIC dynamic fragmented the remaining of the star into disconnected vertices, leaving no further expansion space for the virus. The extinction time τ_e in S_n will be the maximum of I_0 i.i.d. r.v.'s, each with distribution $Exp(\gamma)$, and $\mathbb{E}(\tau_e) = \frac{\mathcal{H}_{I_0}}{\gamma}$, where $\mathcal{H}_k$ is the k-th Harmonic number.

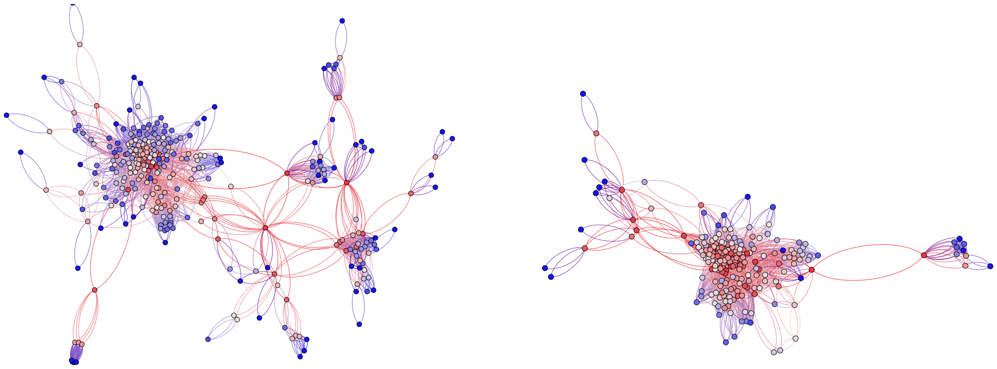
To examine the effectiveness and efficiency of the proposed antidote dissemination policy, we conducted simulation of SIC dynamics on two connected network, both fractions from Facebook [64], `dataset0` [10] and `dataset348`. Statistics of the two networks fraction are shown in Table 2.2. As can be seen from the average degree, `dataset348` is much denser than dataset0. In addition, from the average ccfs (clustering coefficient), `dataset348` is more clustered than `dataset0`.

Figure 2.13 shows the topologies of the two networks, where the betweenness centrality value is indicated by color (low-blue, high-red). Candidates for $\mathcal{C}(0)$ in the betcen-based approach are the vertices in red. Intuitively from the figure, we can tell that `dataset0` is more 'scattered'

[10] Originally, `dataset0` contains 342 vertices and is disconnected, so we select the giant component to be dataset0 during the simulation.

Table 2.2 Statistics of the two networks: `dataset0` and `dataset348`.

Statistics	Description	`dataset0`	`dataset348`
$n = \|\mathcal{V}\|$	order (number of vertices)	324	224
$\|\mathcal{E}\|$	size (number of edges)	5028	6384
$\bar{d} = 2\|\mathcal{E}\|/n$	average degree	31.037	57
$\text{diam}(\mathcal{G})$	network diameter	11	9
$\bar{C}$	average ccfs	0.522	0.544
$\bar{l}$	average path length	3.573	3.042



(a) Topology of `dataset0` network

(b) Topology of `dataset348` network

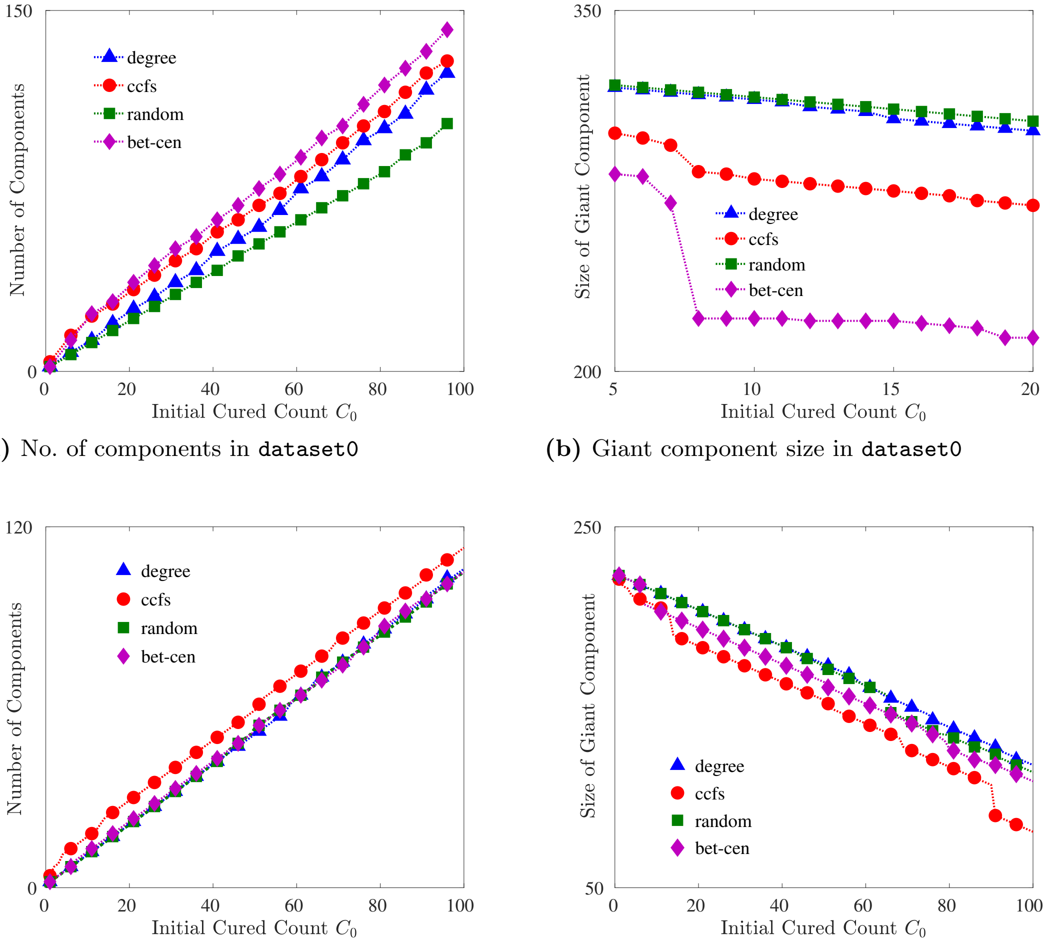
Figure 2.13 Topologies of the two network fractions: Red vertices have high betweenness centrality (betcen, g) values, while blue ones have low g values.

than `dataset348`, which implies that it will be easier (with a smaller $C_0 = |\mathcal{C}(0)|$) to achieve the locking condition, *i.e.* $\mathcal{G}^*(0)$ is disconnected.

Figure 2.14 illustrates topological changes of `dataset0` and `dataset348`, induced by different initial antidote distribution strategies, *i.e.* different assignments of $\mathcal{C}(0)$. These results can be used to predict the effectiveness of those three approaches, plus a random distribution, in terms of $\mathbb{E}(\tau_e)$ and $\mathbb{E}(\tau_{\frac{1}{2}})$. As discussed in Section 2.5.1, the approach that can minimize the size of the giant component will most effectively shorten the extinction time. From the topological characteristics shown in Figure 2.14, it is interesting to see that betcen-based approach will be outperformed by ccfs-based approach in `dataset348`, while in `dataset0` it is the opposite. The possible reason is that `dataset348` is much denser (avg. degree 57) and more clustered (avg. ccfs 0.544) than `dataset0`, which is also manifested in Figure 2.13. This implies the betcen-based approach will disconnect `dataset0` more easily, while leaving `dataset348` still connected during the initial distribution.

Figure 2.15 illustrates the mean extinction time and half-life time, each over 1000 simulation runs. The propagation parameters of the SIC epidemic are: infection rate $\beta = 0.01$, the curing rate $\gamma = 0.01$, and initial infection count $I_0 = |\mathcal{I}(0)| \simeq \frac{1}{2}|\mathcal{V}|$.

Simulation results in Figure 2.15 echoes with the prediction we had from Figure 2.14, in



(a) No. of components in `dataset0`

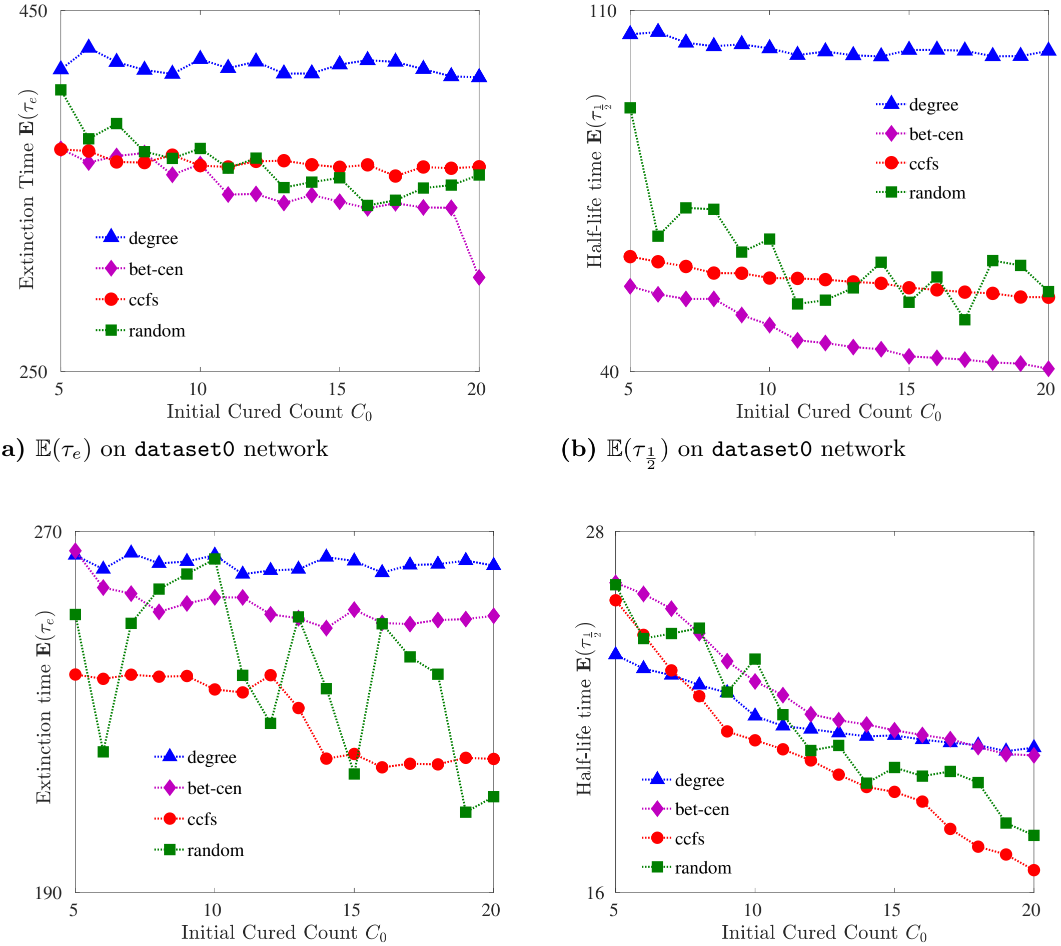
(b) Giant component size in `dataset0`

(c) No. of components in `dataset348`

(d) Giant component size in `dataset348`

Figure 2.14 Topological change of $\mathcal{G}^*(0)$ with under different $\mathcal{C}(0)$ options.

which performance of degree-based approach is poor, and the best approach is either betcen-based or ccfs-based. In terms of extinction time, the degree-based approach is even worse than random distribution, because high degree does not imply high importance. Due to the clustering effect of human social interactions, high degree vertices is often located in a densely connected core, where the removal of such a vertex can be compensated by its neighbors. However, the zig-zag pattern of the green line, *i.e.* the random distribution, indicates the instability of this approach. What's more, considering the ccfs-based approach only requires knowledge of two-hop neighbors, rather than global knowledge in the betcen-based approach, it is an optimal choice, especially when the network is denser. As for the half-life time, which indicates the effectiveness in alleviating heavy infection conditions, simulation suggests similar conclusion as the extinction time, except that degree-based distribution performs better when C_0 is small in `dataset348`, where the network is more clustered.



(a) $\mathbb{E}(\tau_e)$ on `dataset0` network

(b) $\mathbb{E}(\tau_{\frac{1}{2}})$ on `dataset0` network

(c) $\mathbb{E}(\tau_e)$ on `dataset348` network

(d) $\mathbb{E}(\tau_{\frac{1}{2}})$ on `dataset348` network

Figure 2.15 Extinction time $\mathbb{E}(\tau_e)$ and Half-life time $\mathbb{E}(\tau_{\frac{1}{2}})$ under different initial distibution strategies on network `dataset0` and `dataset 348` show that betweeness-centrality based approach works better for `dataset0`, while cluster-coefficient based approach works better for `dataset348`.

2.6 Dynamics in Motion: Estimating the Number of Information Adopters at Time t

In the previous sections, we study the lifetime of the undesired information (virus), which is a 'coarse' and collective way to describe the propagation process at system-level, as only two time pivots are identified from the entire dynamic evolution. To answer the third research question, *how the number of information adopters changes over time*, however, the dynamic process need to be analyzed in finer grains. Therefore, in this section, we discuss the step-by-step evolution of a SIC epidemic process in a discrete time system with time-step of length Δt.

We set Δt to be sufficiently short, such that within a time step, *e.g.*, from the $(t-1)$th-step to tth-step, the state of a vertex $i \in \mathcal{V}$, $X_i(t)$ depends solely on the last states of its neighbors $\{X_j(t-1)|j \in \mathcal{N}(i)\}$ and itself $X_i(t-1)$. This behavior resembles the *Local Markov* *property* of Markov Random Fields (MRF), where (temporal) Markov property is correlated with spatial dependence. To overcome this challenge caused by this correlation, we separate the spatial dependence and temporal dependence through a similar method introduced in [27], but addressing both the virus *infection* and antidote *curing* events at the same time. In this way, we are able to derive a time-recursive expression for the infection probability $\mathbb{P}(X_i(t) = 1)$, from which the expected number of information adopters $\mathbb{E}(I(t))$ and $\mathbb{E}(C(t))$ can be inferred.

Remark 2.4. *Due to the focus on the step-by-step evolution, it is on longer reasonable to use system-level infection/curing rates (also probability, because we adopt Δt as the unit length of a time step) β and γ for all edges. In other words, to obtain a finer time resolution of the dynamics, we also need a finer network model, which can capture the difference between two edges, e.g., the probability $\beta_{i,j}$ that a virus propagate through edge $e(i,j)$ in a time step can be different from $\beta_{u,v}$ that a virus propagate through edge $e(u,v)$. Considering this change in resolution, we write $\beta := \{\beta_{i,j}\}_{i,j\in\mathcal{V}}$ and $\gamma := \{\gamma_{i,j}\}_{i,j\in\mathcal{V}}$, as the matrix of infection probabilities and curing probabilities, respectively. In this sense, the network $\mathcal{G}(\mathcal{V},\mathcal{E})$ is heterogeneous, and can be characterized by matrices β and γ.*

2.6.1 Temporal Dependence

First, we discuss the state dependence and evolution in the time domain. With states of i's neighbors at time t, the probability that a susceptible vertex i remains susceptible in the next time step can be written as

$$\begin{aligned}\mathbb{P}(X_i\left[t+1) = 0 \mid X_i(t) = 0,\ X_{\mathcal{N}(i)}(t) = x_{\mathcal{N}(i)}(t)\right] &= \prod_{j\in\mathcal{N}_I(t,i)} (1-\beta_{j,i}) \cdot \prod_{k\in\mathcal{N}_C(t,i)} (1-\gamma_{k,i}) \\ = \prod_{j\in\mathcal{N}(i)} (1-\beta_{j,i})^{\frac{1}{2}(x_j^2(t)+x_j(t))} \cdot (1-\gamma_{j,i})^{\frac{1}{2}(x_j^2(t)-x_j(t))}. \end{aligned} \tag{2.45}$$

Let $\mathbb{P}_{uv}(t) = \mathbb{P}(X_i(t+1) = v|X_i(t) = u)$ denote the transition probability from state u to v during one time step at time t. Note that only three equations are needed because $\sum_{v\in\Lambda} \mathbb{P}_{u,v} = 1$, $\mathbb{P}_{10} = 0$ and $\mathbb{P}_{-1-1} = 1$. Let $I_i(t) = 1 - \prod_{j\in\mathcal{N}(i)}(1-\beta_{j,i})^{\frac{1}{2}(x_j^2(t)+x_j(t))}$, $C_i(t) = 1 - \prod_{k\in\mathcal{N}(i)}(1-\gamma_{k,i})^{\frac{1}{2}(x_k^2(t)-x_k(t))}$, then the evolution can be described by the following one-step transition probabilities.

$$\mathbb{P}_{00}(i,t) = \sum_{x_{\mathcal{N}(i)}(t)} \mathbb{P}(X_{\mathcal{N}(i)}(t) = x_{\mathcal{N}(i)}(t)|X_i(t) = 0) \cdot (1-I_i(t))(1-C_i(t)), \tag{2.46}$$

$$\mathbb{P}_{01}(i,t) = \sum_{x_{\mathcal{N}(i)}(t)} \mathbb{P}(X_{\mathcal{N}(i)}(t) = x_{\mathcal{N}(i)}(t)|X_i(t) = 0) I_i(t)(1-C_i(t)), \tag{2.47}$$

$$\mathbb{P}_{11}(i,t) = \sum_{x_{\mathcal{N}(i)}(t)} \mathbb{P}(X_{\mathcal{N}(i)}(t) = x_{\mathcal{N}(i)}(t)|X_i(t) = 1) C_i(t). \tag{2.48}$$

The temporal dependence is included in $I_i(t)$ and $C_i(t)$, while the spatial dependence is captured by the joint conditional probability $\mathbb{P}(X_{\mathcal{N}(i)}(t) = x_{\mathcal{N}(i)}(t)|X_i(t) = x_i(t))$, which is the reason that calculating the exact marginal probability distribution is expensive. A common assumption is that during one time step, states of different vertices are mutually independent, *i.e.* $\mathbb{P}(X_{\mathcal{N}(i)}(t) = x_{\mathcal{N}(i)}(t)|X_i(t) = x_i(t)) = \prod_{j\in\mathcal{N}(i)} \mathbb{P}(X_j(t) = x_j(t))$, which creates a noticeable deviation in probability as indicated by [27], so we discuss spatial dependence in the next step.

2.6.2 Spatial Dependence

The relationship between the temporal and spatial state dependence is that, during one time step, *i.e.*, from t to $t+1$, virus and antidote can only propagate to vertices that are at most one hop away, or equivalently, only the states of one's neighbors are relevant in determining the next state of a vertex. In other words, we can write the temporal and spatial dependence as follows:

$$X_u(t+1) \ \perp \ X_{\mathcal{V}\backslash\mathcal{N}(u)\cup\{u\}}(t) \mid X_{\mathcal{N}(u)}(t). \tag{2.49}$$

Based on this observation, we assume the states of vertex i's neighbors $\mathcal{N}(i)(t)$ are independent of each other for every time step t, so

$$\mathbb{P}(X_{\mathcal{N}(i)}(t) = x_{\mathcal{N}(i)}(t)|X_i(t) = x_i(t)) = \prod_{j\in\mathcal{N}(i)} \mathbb{P}(X_j(t) = x_j(t)|X_i(t) = x_i(t)). \tag{2.50}$$

Let $N_0^s(i,j,t) = \mathbb{P}(X_j(t) = s|X_i(t) = 0)$, $N_1^s(i,j,t) = \mathbb{P}(X_j(t) = s|X_i(t) = 1)$, $s \in \{0,-1,1\}$. Then Eq. (2.46), (2.47), and (2.48) can be simplified into

$$\begin{aligned}
\mathbb{P}_{00}(i,t) &= \sum_{x_{\mathcal{N}(i)}(t)} \prod_{j\in\mathcal{N}(i)} \left[N_0^{x_j(t)}(i,j,t) \cdot (1-\beta_{j,i})^{\frac{1}{2}(x_j^2(t)+x_j(t))} \cdot (1-\gamma_{j,i})^{\frac{1}{2}(x_j^2(t)-x_j(t))} N_0^{x_j(t)}(i,j,t)\right] \\
&= \prod_{j\in\mathcal{N}(i)} \left[1 - N_0^1(i,j,t)\beta_{j,i} - \big(1 - N_0^0(i,j,t) - N_0^1(i,j,t)\big)\gamma_{j,i}\right] &(2.51)\\
\mathbb{P}_{01}(i,t) &= \sum_{x_{\mathcal{N}(i)}(t)} \Big[1 - \prod_{j\in\mathcal{N}(i)} (1-\beta_{j,i})^{\frac{1}{2}(x_j^2(t)+x_j(t))}\Big] \cdot \prod_{j\in\mathcal{N}(i)} (1-\gamma_{j,i})^{\frac{1}{2}(x_j^2(t)-x_j(t))} N_0^{x_j(t)}(i,j,t) \\
&= \prod_{j\in\mathcal{N}(i)} \left[1 - \big(1 - N_0^0(i,j,t) - N_0^1(i,j,t)\big)\gamma_{j,i}\right] - \mathbb{P}_{00}(i,t) &(2.52)\\
\mathbb{P}_{11}(i,t) &= 1 - \sum_{x_{\mathcal{N}(i)}(t)} \Big[1 - \prod_{j\in\mathcal{N}(i)} (1-\gamma_{j,i})^{\frac{1}{2}(x_j^2(t)-x_j(t))}\Big] \cdot \prod_{j\in\mathcal{N}(i)} N_0^{x_j(t)}(i,j,t) \\
&= \prod_{j\in\mathcal{N}(i)} \left[1 - \big(1 - N_1^0(i,j,t) - N_1^1(i,j,t)\big)\gamma_{j,i}\right]. &(2.53)
\end{aligned}$$

With Eq. (2.51)-(2.53), the recursive master equations of the system over time can be written as

$$\mathbb{P}(X_i(t+1) = 0) = \mathbb{P}(X_i(t) = 0) \cdot \mathbb{P}_{00}(i,t), \tag{2.54}$$

$$\mathbb{P}(X_i(t+1) = 1) = \mathbb{P}(X_i(t) = 0) \cdot \mathbb{P}_{01}(t) + \mathbb{P}(X_i(t) = 1) \cdot \mathbb{P}_{11}(i,t). \tag{2.55}$$

Note $N_0^0(i,j,t), N_0^1(i,j,t), N_1^0(i,j,t)$ and $N_1^1(i,j,t)$ are necessary to solve these equations, but it is hard to obtain closed form equations of these quantities. However, it is possible to derive recursive expressions for them with respect to time, as shown below.

$$\begin{aligned}
N_0^0(i,j,t) &= \frac{\mathbb{P}(X_j(t)=0, X_i(t)=0)}{\mathbb{P}(X_i(t)=0)} \\
&= \frac{\mathbb{P}(X_i(t-1)=0)}{\mathbb{P}(X_i(t)=0)} \cdot N_0^0(i,j,t-1) \cdot \mathbb{P}_{00}(i,j,t-1) \cdot \mathbb{P}_{00}(j,i,t-1), \qquad (2.56) \\
N_0^1(i,j,t) &= \frac{\mathbb{P}(X_j(t)=1, X_i(t)=0)}{\mathbb{P}(X_i(t)=0)} \\
&= \frac{\mathbb{P}(X_i(t-1)=0)}{\mathbb{P}(X_i(t)=0)} \cdot P_{00}(i,j,t-1) \cdot \big[N_0^0(i,j,t-1) \cdot \mathbb{P}_{01}(j,i,t-1) \\
&\quad + (1-\beta_{j,i}) \cdot N_0^1(i,j,t-1) \cdot \mathbb{P}_{11}(j,i,t-1)\big], \qquad (2.57) \\
N_1^0(i,j,t) &= \frac{\mathbb{P}(X_j(t)=0, X_i(t)=1)}{\mathbb{P}(X_i(t)=1)} \\
&= \frac{\mathbb{P}_{00}(j,i,t-1)}{\mathbb{P}(X_i(t)=1)} \cdot \big[\mathbb{P}(X_i(t-1)=0) \cdot N_0^0(i,j,t-1) \cdot \mathbb{P}_{01}(i,j,t-1) \\
&\quad + \mathbb{P}(X_i(t-1)=1) \cdot (1-\beta_{i,j}) \cdot N_1^0(i,j,t-1) \cdot \mathbb{P}_{11}(i,j,t-1)\big], \qquad (2.58) \\
N_1^1(i,j,t) &= \frac{\mathbb{P}(X_j(t)=1, X_i(t)=1)}{\mathbb{P}(X_i(t)=1)} \\
&= \frac{\mathbb{P}(X_i(t-1)=0)}{\mathbb{P}(X_i(t)=1)} \cdot \Big[N_0^0(i,j,t-1) \cdot \mathbb{P}_{01}(i,j,t-1) \cdot \mathbb{P}_{01}(j,i,t-1) \\
&\quad + N_0^1(i,j,t-1) \cdot \mathbb{P}_{11}(j,i,t-1) \cdot [\beta_{j,i} + \mathbb{P}_{01}(i,j,t-1) - \beta_{j,i}\mathbb{P}_{01}(i,j,t-1)]\Big] \\
&\quad + \frac{\mathbb{P}(X_i(t-1)=1)}{\mathbb{P}(X_i(t)=1)} \cdot \mathbb{P}_{11}(j,i,t-1) \cdot \Big[[\beta_{i,j} + \mathbb{P}_{01}(j,i,t-1) - \beta_{i,j}\mathbb{P}_{01}(j,i,t-1)] \\
&\quad \cdot N_1^0(i,j,t-1) + N_1^1(i,j,t-1) \cdot \mathbb{P}_{11}(i,j,t-1)\Big], \qquad (2.59)
\end{aligned}$$

where transition probabilities $\mathbb{P}_{00}(i,j,t)$, $\mathbb{P}_{01}(i,j,t)$, and $\mathbb{P}_{11}(i,j,t)$

$$\mathbb{P}_{00}(i,j,t) = \prod_{k \in \mathcal{N}(i) \backslash \{j\}} \big[1 - N_0^1(i,k,t)\beta_{k,i} - \big(1 - N_0^0(i,k,t) - N_0^1(i,k,t)\big)\gamma_{k,i}\big] \tag{2.60}$$

$$\mathbb{P}_{01}(i,j,t) = \prod_{k \in \mathcal{N}(i) \backslash \{j\}} \big[1 - \big(1 - N_0^0(i,k,t) - N_0^1(i,k,t)\big)\gamma_{k,i}\big] - P_{00}(i,j,t) \tag{2.61}$$

$$\mathbb{P}_{11}(i,j,t) = \prod_{k \in \mathcal{N}(i) \backslash \{j\}} \big[1 - \big(1 - N_1^0(i,k,t) - N_1^1(i,k,t)\big)\gamma_{k,i}\big]. \tag{2.62}$$

denote the probability that without considering vertex $j \in \mathcal{N}(i)$, vertex i remains susceptible, becomes infected, and remains infected during time step t respectively.

2.6.3 Expected Infection Count $\mathbb{E}(I(t))$ and Cured Count $\mathbb{E}(C(t))$

Note that Eq. (2.56)-(2.59) are time-recursive, allowing us to estimate the state evolution as long as the initial state of the network is known (equivalent to the case that the initial distribution of state vector $\mathbf{X}(0)$ is a δ-distribution at time $t = 0$). It is clear that $\mathbb{P}(X_v(0) = 1) = 1\ \forall v \in \mathcal{I}(0)$, $\mathbb{P}(X_u(0) = -1) = 1\ \forall u \in \mathcal{C}(0)$ and $\mathbb{P}(X_w(0) = 0) = 1\ \forall w \in \mathcal{S}(0)$. Therefore, $\mathbb{P}(\mathbf{X}(0) = \mathbf{x}(0)) = 1 = \prod_{v\in\mathcal{V}} \mathbb{P}(X_v(0) = x_v(0))$, where $\mathbf{x}(0) = \{x_v(0)\}_{v\in\mathcal{V}}$ is the state vector of the network at time 0, which indicates r.v.'s $\{X_v(0)\}_{v\in\mathcal{V}}$ are mutually independent. Hence $N_r^s(j,0) = \mathbb{P}(X_j(0) = s|X_i(0) = r) = \mathbb{P}(X_j(0) = s)$ is determined for any $r, s \in \{0, 1, -1\}$. Then $\mathbb{P}(X_i(t) = 1)$ can be solved iteratively, and the expected infection count at time t can be calculated by

$$\mathbb{E}(I(t)) = \mathbb{E}[\sum_{i\in\mathcal{V}} \mathbb{1}_{\{1\}}(X_i(t))] = \sum_{i\in\mathcal{V}} \mathbb{P}(X_i(t) = 1), \tag{2.63}$$

$$\mathbb{E}(C(t)) = \sum_{i\in\mathcal{V}} \big[1 - \mathbb{P}(X_i(t) = 0) - \mathbb{P}(X_i(t) = 1)\big]. \tag{2.64}$$

The manipulations discussed above, whose purposes are to obtain $\mathbb{E}(I(t))$ and $\mathbb{E}(C(t))$ for a given time step t, can be summarized into the following iterative algorithm. Note that $\vec{P}_0(t) = \{\mathcal{P}(X_i(t) = 0)\}_{i\in\mathcal{V}}$ and $\vec{P}_1(t) = \{\mathcal{P}(X_i(t) = 1)\}_{i\in\mathcal{V}}$ are two $n \times 1$ vectors, while $\vec{RS}(t) = \{\mathbb{P}_{00}(i,j,t)\}_{i,j\in\mathcal{V}}$, $\vec{NC}(t) = \{\mathbb{P}_{11}(i,j,t)\}_{i,j\in\mathcal{V}}$, $\vec{N}_0^0(t) = \{N_0^0(i,j,t)\}_{i,j\in\mathcal{V}}$, $\vec{N}_1^0(t) = \{N_0^1(i,j,t)\}_{i,j\in\mathcal{V}}$, $\vec{N}_0^1(t) = \{N_1^0(i,j,t)\}_{i,j\in\mathcal{V}}$,$\vec{N}_1^1(t) = \{N_1^1(i,j,t)\}_{i,j\in\mathcal{V}}$ are $n \times n$ matrices.

Algorithm 1 Iterative Inference Method.

```
Input: $\mathcal{I}(0)$, $\mathcal{C}(0)$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, $\mathcal{V}$
Output: $\mathbb{E}(I(t))$, $\mathbb{E}(C(t))$
Initialize: $n = |\mathcal{V}|$, $I(0) = |\mathcal{I}(0)|$, $C(0) = n - I(0) - |\mathcal{C}(0)|$
            $\vec{P}_0(0) = \vec{N}_0^0(0) = \vec{N}_1^0(0) = \mathbb{1}_{\mathcal{V}\setminus(\mathcal{I}(0)\cup\mathcal{C}(0))}$,
            $\vec{P}_1(0) = \vec{N}_0^1(0) = \vec{N}_1^1(0) = \mathbb{1}_{\mathcal{I}(0)}$
Calculate $\vec{P}_0(1)$ with Eq. (2.48) and (2.53)
Calculate $\vec{P}_1(1)$ with Eq. (2.51), (2.52) and (2.55)
Output $\mathbb{E}(I(1)) = sum(\vec{P}_1(1))$,
       $\mathbb{E}(C(1)) = n - \mathbb{E}(I(1)) - sum(\vec{P}_0(1))$
Calculate $\vec{RS}(0)$ and $\vec{NC}(0)$ with Eq. (2.60) and (2.61)
$t = 1$
while $\mathbb{E}(C(t)) < n$ do
    Calculate $\vec{N}_0^0(t)$, $\vec{N}_0^1(t)$, $\vec{N}_1^0(t)$, $\vec{N}_1^1(t)$ with Eq. (2.56)-(2.60)
    Calculate $\vec{P}_0(t+1)$ with Eq. (2.48) and (2.53)
    Calculate $\vec{P}_1(t+1)$ with Eq. (2.51), (2.52) and (2.55)
    Output $\mathbb{E}(I(t+1)) = sum(\vec{P}_1(t+1))$,
           $\mathbb{E}(C(1)) = n - \mathbb{E}(I(t+1)) - sum(\vec{P}_0(t+1))$
    Calculate $\vec{RS}(t)$ and $\vec{NC}(t)$ with Eq. (2.60) and (2.61)
    $t = t + 1$
end while
```

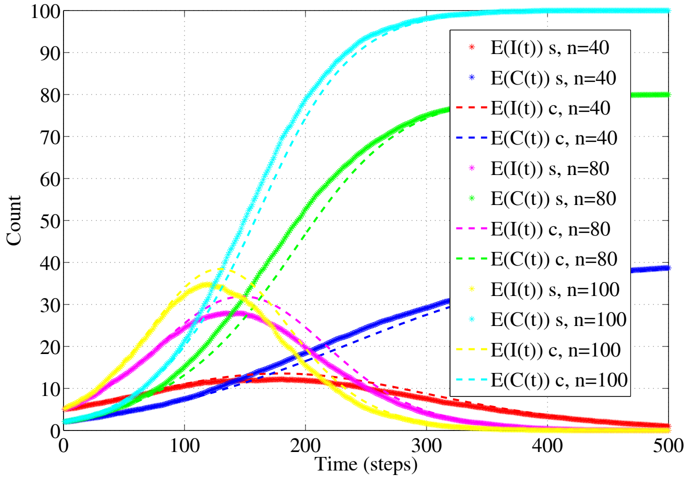


Figure 2.16 Simulated *v.s.* calculated infection/cured count in networks of size 40,80 and 100.

We validate the efficacy of the proposed algorithm in networks with size n ranging from 40 to 100, where each edge $e(i, j)$ is associated with random infection/curing probability $\beta_{i,j}$ and $\gamma_{i,j}$. Figure 2.16 shows the simulation (dashed lines with s in the legend) v.s. calculation (solid lines composed of dots, with c in the legend) of expected value of infection and cured counts. The infection count $I(t)$ and the curing count $C(t)$ can be viewed as the number of information adopters at time t. As shown by the small gaps between the solid lines and their corresponding dashed lines, $\mathbb{E}(I(t))$ and $\mathbb{E}(C(t))$ can be well captured by the outcome of Algorithm 1.

2.7 Summary

In this chapter, we studied conflicting information propagation, in terms of lifetime of the undesired information. We propose a Susceptible-Infected-Cured propagation model to capture the short-term competitions between the virus (undesired information) and the later-injected antidote (undesired information), both propagating on the same network. We find the lifetime of the virus can be upper bounded by two topological properties of the network $\mathcal{G}$: the Cheeger constant $\eta(\mathcal{G})$, which measures the level of 'bottleneckness', and vertex eccentricities $\{\epsilon_{\mathcal{G}}(v)\}_v$, which measures the longest distance between vertex v and any other vertices in the network. Particularly, the $O(\frac{\log n}{\eta(\mathcal{G})})$ bound indicates that the lifetime of the virus does not always decrease with network size n, while the upper bounds with $\{\epsilon_{\mathcal{G}}(v)\}_v$ are less computationally expensive to obtain. As an application of this knowledge, we provide practical antidote distribution strategies with real-world network traces. Finally, we propose an inference algorithm to estimate the number of information adopters, which can foresee the instantaneous evolution of the dynamics before it fully unfolds. As the driving force of mobile data, information dynamics determines *when* data moves and stops, as well as the volume of data traffic in networks. Knowledge on the lifetime of information provides guidelines on network topology design, data traffic prediction, and control of undesired information propagation.

Chapter 3

Coverage Dynamics: Modeling, Analysis, and Prediction of Data Coverage in Heterogeneous Edge Networks

Our study in the previous chapter revealed the *lifetime* of mobile data in large networks, to address the *when* question. In this chapter, we shift gear to another important aspect of mobile data in the space domain, that is, the *whereabouts* of data. Specifically, we study *where* data are accessible, *i.e.*, the *data coverage*, during a dissemination process. This question is of particular importance to location-based service (LBS), in which data are tagged with geographical locations, and disseminated though mobile devices with diverse capabilities in the edge network. However, existing research on data dissemination either do not carry the notation of 'where', or assume participants of the dissemination process to be homogeneous in communication, mobility, and other aspects. To overcome these challenges that hinder the understanding of data dissemination services, we propose a data coverage model that captures the attributes and actions of individual participants, and quantify the data coverage with a *data strength* metric, by which the dynamic change of data coverage can be described as a numeric signal on a *graph*, that embeds location information. Analyzing this graph signal over entity mobility and data forwarding, we find that, the impact of entity mobility is high especially when entities are less active in forwarding, or moving in low speed, and such impact acts in the form of a user mobility matrix, which describes the underlying graph topology of the defined signal of data strength. This observation enables us to *predict* future data coverage with graph signal processing (GSP) tools. Our results provide a novel, yet practical solution to understand data dissemination services in the edge network composed of heterogeneous wireless devices.

3.1 Introduction

Recent advances in data-driven applications are re-enforcing the indispensable role of rapidly growing *mobile data* [120] in a variety of scenarios, ranging from marketing [126] to manufacturing [114], and scientific research [40]. In the era of mobile data, generation, dissemination and consumption of data are much easier than ever, owing to the proliferation of increasingly powerful mobile devices, such as smart phones [142], as well as emerging networking paradigms, such as IoT and fog computing [85, 18]. Consequently, data dissemination processes, which move data from their generator(s) to their consumer(s), as if data are a special type of commodities, are quickly migrating to the *edge* networks, which are composed of both user-operated wireless devices that can move, such as smart phones, wearable devices, and smart vehicles; and stationary network elements that provide wireless access to devices as a part of the infrastructure, such as cellular BS, LTE eNodeB, and WiFi AP.

3.1.1 Motivation

Such data dissemination services are imperative to a lot of applications. For instance, in mobile advertisement (ad) applications [99, 124], business-owners can recruit smart phones, nearby vehicles and AP's to distribute ads, such that an ad penetration is maintained over populous regions. Another example is traffic information collection/distribution in intelligent transportation systems (ITS) [130, 8], which heavily relies on communication among smart vehicles with on-board unit (OBU), IoT sensors, RSU, and cellular eNodeB. Despite their differences, data dissemination processes in these applications share some common grounds:

(1) *Heterogeneous network*: Data are generated and disseminated by mobile phones, smart vehicles, AP's, and BS's, which we refer to as *entities*, with vastly different communication schemes, mobility patterns, and forwarding preferences. For example, the communication range of a BS (100 m to 10 Km) is much larger than that of an AP (20 m to 1 Km), and a pedestrian, who streams video over his/her smart phone, moves much slower than a vehicle on the highway, whose registration information is collected and routed by the electric toll collecting (ETC) sensors. As a result, entities may intermittently connect, disconnect, and re-connect as they move, so data to be disseminated traverse a heterogeneous network via both wired (*e.g.*, the X2 connection between LTE eNodeB's and wireless (*e.g.*, the D2D connection between LTE smart phones) links, before they are picked up by consumers.

(2) *Mobile data*: Ever since its injection into the network edge, a data block (which we refer to as *datum* throughout this chapter) and its copies are constantly *moving*, due to user/entity movements such as people walking, data forwarding actions such as ad broadcasting, and entities' decisions to participate in/withdraw from the dissemination process.

(3) *Location-based service (LBS)*: The dissemination service is location-centric, because *where* data are determines *who* can immediately access the data. This is especially evident to applications that deal with location-tagged data, *e.g.*, traffic information collected and ex-

changed among smart vehicles, mobile ads distributed by local stores in a mall, sensory data collected and actuation commands distributed in IoT, etc.

A key question arises from the dissemination process of data: *where are the accessible data*?

Answer to the 'where' question is meaningful to multiple stake-holders of a data dissemination service, and can effectively guide a number of design issues, such as placement of network elements, allocation of storage/processing resources, and evaluation of network performances. From the perspective of data owners, *e.g.*, companies who purchase mobile ad service from AdMob, it is their legitimate rights to know where their data can be accessed, and preferably accessed by whom. To data service providers, *e.g.*,content delivery network (CDN) like Cloudflare, whereabouts of data should be disclosed to content owners for service transparency, and as input to servers deployment planning. In regard to the network operators, *e.g.*, cellular carriers like AT&T, changing volume of mobile data translate to traffic engineering, and management of network resources, which should be considered in network planning, charging, and OA&M (operation, administration, and maintenance). Therefore, it is necessary for all parties, to have a quantitative measure on the *whereabouts* of data in dissemination, and preferably a mechanism to evaluate/predict data accessibility over a geographical region of interest.

3.1.2 Related Work

Due to its great importance, data dissemination process in wireless networks has been extensive studied from the following aspects, such as data dissemination time [141], throughput [46], and network topology [99]. Among these, Grossglaussor and Tse proved movements of entities (under speed-constrained mobility) can increase the system capacity (in terms of pairwise throughput) for ad hoc networks [46] relying uni-casts. Under a move-and-gossip dissemination model, Zhang *et.al.* defined a mobile conductance metric, to derive the spreading time, by when all of the entities have received the data [141]. Li *et.al.* studied the speed of data propagation with geo-cast [66] in a group of mobile devices following the same propagation mode and mobility pattern. Lu *et.al.* showed there exists a critical condition characterized by participating probability, moving speed and transmission range, above which the number of data-carrying entities will scale quadratically with time [76]. In the aforementioned literature, participants of the data dissemination process are assumed to be homogeneous in aspects including communication capacity, mobility pattern, and data forwarding preferences. This is not the case in the edge network, where participants can be vastly different in these aspects. Taking heterogeneity in communication protocols into consideration, hierarchical topology of mobiles in a cellular and WiFi overlay network is studied in [68]. From the perspective of data forwarding influenced by social features, Qin *et.al.* analyzed the dynamic sociality of vehicles, based on which they proposed a mobile advertisement dissemination scheme that can achieve a shorter distributing time [100]. Despite it significant impact, the 'where' question remains open, as the aforementioned literature do not carry the notion of data's whereabouts.

3.1.3 Our Approach and Contributions

Motivated by the lack of study on the whereabouts of data, this chapter discusses *data coverage*, that is, the time-varying locations where a data block of interest can be accessed during its dissemination process. However, answer to the 'where' question are hindered by two main challenges arsing from the edge network scenario: First, from the unique setting of data dissemination processes, how to account for the heterogeneity among entities, including both mobile devices, and stationary access network elements that are different in many aspects? Second, from the location-centric nature of the problem, how to quantitatively describe the whereabouts of a data block (and its copies), considering that they are constantly moving over-the-air due to forwarding, being downloaded and/or deleted by users, and piggybacking on moving entities?

To overcome the first challenge of heterogeneity, we propose a generic entity model that captures the attributes and actions of participants in a dissemination process, and formally define a measurable scope of *data coverage*, *i.e.*, the geographical region in which the data block can be accessed to answer the 'where' question. Then to address the second challenge of quantification, we define a new metric, *data-strength*, through which the *coverage dynamics*, *i.e.*, time-varying data coverage, can be quantitatively described by a signal on a *graph* that embeds geographical location information. The graph signal formulation allows us to study the evolution and prediction of the dynamic process of data movement and coverage. Particularly, we analyze the impact of user mobility and data forwarding through a *snapshot* of the coverage dynamics, *i.e.*, the difference graph signal. Based on the observation and the governing equations of the dynamics, we propose and test a graph signal processing (GSP)-based prediction framework, to showcase an application of the proposed model. In summary, this chapter makes the following contributions toward understanding mobile data dissemination in the edge networks:

We formally define the data movement and coverage problem, propose a new metric, data-strength, to characterize the dissemination process, and formulate data coverage as a tractable time series of numeric graph signals.

We observe bounded impact from user mobility and data forwarding on the change of data-strength, as the maximum moving speed or participating ratio increases. Particularly, the impact of user mobility is high when entities are not active in moving or data forwarding, and this impact takes effect in the form of a mobility matrix, which describes the underlying graph of the data strength signal. It indicates that data coverage in the future can be predicted based on the current data strength, and the long-term mobility patterns.

We propose and implement a GSP-based prediction framework, which can predict the instantaneous data coverage with an over 80% accuracy for time periods ranging from 10 s to 200 s, in data dissemination scenarios with both wireless and wired connections.

3.2 Problem Formulation

As discussed in the previous section, existing research on data dissemination focus on 'how many' have received the data, instead of 'where' data can be accessed. To introduce the notion of data whereabouts, we introduce the *data coverage* model in this section, to formulate the intuitive 'where' question into specific and rigorous research questions. Specifically, we first specify the scope of the 'where' question, establish the entity model to characterize dissemination participants of various kinds, and formally define data coverage dynamics, based on which the 'where' question is break into in three tractable sub-questions.

3.2.1 Scope of the 'Where' Problem

The time-varying whereabouts of data due to entity movements and data forwarding are relevant to three domains: time, space, and data, for each of which we specify our assumptions.

(1) *Time*: The system runs in continuous time $[0, \infty)$, and is observed every Δ time. The 0-th observation taken at time instant 0 is called the initial state of the system. For integers $t \in \mathcal{T} = \{1, 2, \cdots\}$, we differentiate the following:

- Time (instant) t refers to the time *instant* $t \cdot \Delta \in [0, \infty)$, when we take the t-th observation of the dissemination process.

- Time step t refers to the time *interval* $[(t-1)\Delta, t\Delta) \subset [0, \infty)$, during which actions and interactions of entities take place in the system.

With this clarification, we consider intervals of unit length ($\Delta = 1$), and write $t \in \mathcal{T}$ throughout this chapter, when no confusion is raised.

(2) *Space*: Consider a planar region $A \subset \mathbb{R}^2$, *e.g.*, the downtown area of a city, where mobile ads are disseminated through smart phones. Let $d_A(\cdot, \cdot)$ denote the Euclidean distance in A.

(3) *Data*: Let D denote a set of distinct data blocks. We consider the dissemination process of one data block $\delta \in D$, later referred to as a *datum*, which has been injected into the network before time instant $t = 0$. The dissemination process of datum δ refers to the replication, forwarding, and deletion of δ that take place in region A. Exact copies of datum δ (as a result of replication) are treated as the same data block as the original datum δ during the dissemination process, but any distorted, or altered copies of δ are viewed as different datum than δ, and is hence not considered in the dissemination process.

Participants of a data dissemination process include mobile wireless devices, *e.g.*, smart phones, IoT sensors, and OBU's in vehicles, as well as infrastructure nodes in wireless access networks, *e.g.*, AP's, eNodeB's, and RSU's. They compose a group of *entities* E, each of which can receive, carry, distribute, and delete datum δ at any time.

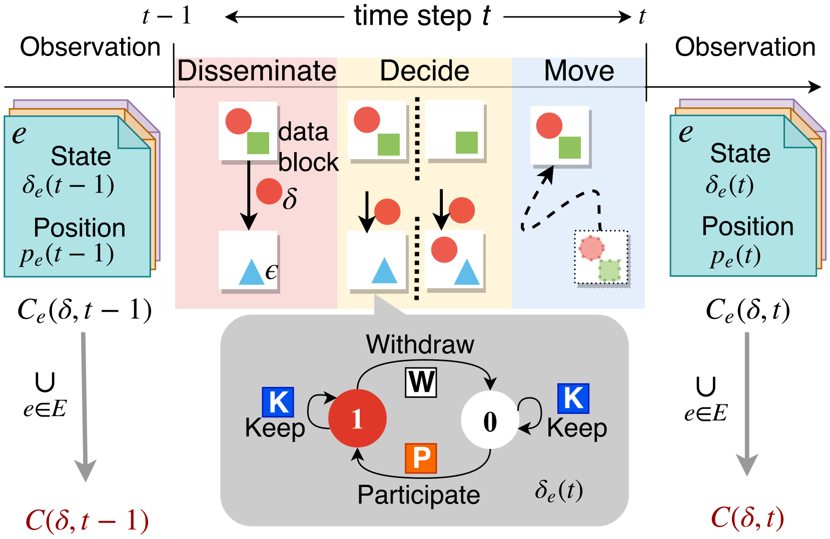


Figure 3.1 Data coverage model (Note that time is not to scale in this diagram).

3.2.2 Entity Model

Each entity $e \in E$ is characterized by a list of attributes and a set of actions that can change entity e's own dynamic attributes (and hence the whereabouts of datum δ).

3.2.2.1 Attributes of Entity e

Attributes of an entity describe its (static) capability and (dynamic) statuses.

(1) *Transmission range* $r_e \in \mathbb{R}^+$ is defined as the maximum distance that a data block $\delta \in D$ can be transmitted successfully over wireless links by entity e in one time step. It is a static attribute that does not change over time.

(2) *Position* $p_e(t) \in \mathbb{R}^2$ records the exact location, *e.g.*, GPS coordinates, of entity e at every observation, which is taken at time t. It is a dynamic status, and note that it is possible for an entity e to move out of (and back into) region A at some time instant.

(3) *State* $\delta_e(t) \in \{0, 1\}$ is an indicator of whether entity e has a copy of datum δ at time instant t, as well as a determinant of entity e's actions in the next ($t + 1$-th) time step. Entity e will participate in the dissemination process, *i.e.*, promises to disseminate copies of δ, during time step $t + 1$, if and only if its state $\delta_e(t) = 1$. It is a dynamic status of entity e.

The dynamic statuses of an entity e, that is, position $p_e(t)$ and states $\delta_e(t)$ observed at time instant t, are changed by actions the entity takes in past time step t (as time instant t marks the end of time step t, as shown in Figure 3.1), and the current statuses partially determine the actions entity e will take during time step $t + 1$.

3.2.2.2 Actions

During every time step $t \geq 1$, each entity takes three actions sequentially: one disseminate action, one decide action, and a move action, as illustrated in the central block of Figure 3.1.

(1) *Disseminate*: At the beginning of time step t, given its previous state $\delta_e(t-1)$, entity e will take exactly one of the following two actions:

- *Forward* datum δ to all possible recipients, if entity e decided to participate (in the previous time step), *i.e.*, $\delta_e(t-1) = 1$. Data recipients include any entity $x \in E$ within its wireless communication range, that is, $\{x \in E \mid d_A(p_e(t), p_x(t)) \leq \min\{r_c^e, r_c^x\}\}$, and (stationary) entity y that is directly connected with (stationary) entity e via wired links;
- *Listen* to broadcasts of others, if $\delta_e(t-1) = 0$.

(2) *Decide*: Then entity e makes a decision to set its state $\delta_e(t)$, *i.e.*, whether entity e will:

- *Keep* its current state during time step $t+1$, such that $\delta_e(t) = \delta_e(t-1)$, as shown by the blue boxes with letter K in Figure 3.1;
- *Withdraw* from the dissemination process, by deleting the local copy of δ and setting $\delta_e(t) = 0$, given that it was participating in the previous time step, *i.e.*, $\delta_e(t-1) = 1$, as shown by the white box with letter W in Figure 3.1;
- *Participate* in the dissemination process of datum δ, by setting state $\delta_e(t) = 1$, as shown by the orange box with letter P in Figure 3.1, given that i) it was not participating in the last time step, *i.e.*, $\delta_e(t-1) = 0$, and ii) it has received a copy of δ during the dissemination stage in time step t.

(3) *Move*: Finally entity e moves from its last position $p_e(t-1)$ to a new position $p_e(t) \in \mathbb{R}^2$. Though stationary entities like an eNodeB are stationary, we write $p_e(t) = p_e(t-1)$, $\forall\ t \in \mathcal{T}$, such that all entities can be described by the same generic model.

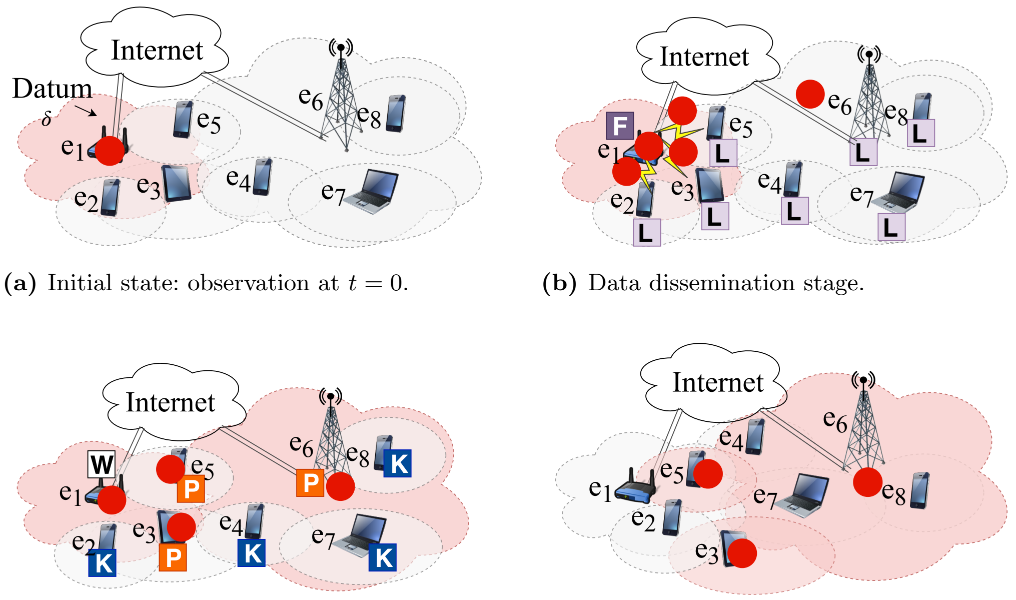
3.2.3 Data Coverage and Coverage Dynamics

Due to the movements, actions, and interactions of entities in E, the region where a datum can be accessed, *i.e.*, its data coverage, changes over time.

Definition 3.1. ***(Data) Coverage*** $C(\delta, t)$ ***of datum*** *δ at time t is defined as a sub-region of region A, where a copy of δ can be obtained in time step $t+1$, that is,*

$$C(\delta, t) := \{a \in A \mid \exists\ e \in E,\ s.t.\ a \in C_e(\delta, t)\}, \tag{3.1}$$

where $C_e(\delta, t)$ denotes the individual coverage offered by entity e, i.e., locations where δ can be



(a) Initial state: observation at $t = 0$.

(b) Data dissemination stage.

(c) Individual decision stage.

(d) Next state: observation at $t = 1$.

Figure 3.2 An illustration of data movement and coverage (red shaded area) during interval $[1, 2]$, in a heterogeneous edge network composed of eight entities $\{e_i\}_{i\in[1,8]}$, two of which (AP e_1 and eNodeB e_6) are stationary entities connected by wired links.

obtained from e, that is,

$$C_e(\delta, t) := \begin{cases} \{a \in A \mid d_A(a, p_e(t)) \le r_e\}, & \textit{if } \delta_e(t) = 1, \\ \phi, & \textit{otherwise.} \end{cases} \tag{3.2}$$

An Example. Figure 3.2 shows the time-varying data coverage in a system of eight entities, two (AP e_1 and eNodeB e_6) of which are stationary entities. During this (short) dissemination process, the system is observed at time instant $t = 0$ and $t = 1$, as shown in Figure 3.2a and d. During time step 1, the following individual actions drive the coverage dynamics, *i.e.*, changes the data coverage:

(a) By time instant $t = 0$, datum δ has been injected into the system through entity e_1, so $\delta_{e_1}(0) = 1$. Data coverage $C(\delta, 0) = C_{e_1}(\delta, 0)$, and $C_{e_i}(\delta, 0) = \phi$ for $i \neq 1$.

(b) At the beginning of time step 1 (dissemination stage), entity e_1 forward datum δ to potential recipients $\{e_2, e_3, e_5, e_6\}$, while all other entities listen, as shown by the light purple boxes with letter L. Note a copy of δ is passed to entity e_6 via the wired link.

(c) Then in the decision stage, entity e_3, e_5, e_6 decide to participate ($\delta_{e_3}(1) = \delta_{e_5}(1) = \delta_{e_6}(1) = 1$); entity e_2, e_4, e_7 and e_8 decide to keep their current states ($\delta_{e_2}(1) = \delta_{e_4}(1) = \delta_{e_7}(1) = \delta_{e_8}(1) = 0$); entity e_1 decides to withdraw from the dissemination ($\delta_{e_1}(1) = 0$).

(d) At time instant $t = 1$, the system is observed again (after each mobile entity moves) for data coverage $C(\delta, 1)$, which equals to $\cup_{i=3,5,6} C_{e_i}(\delta, 1)$.

In this way, the coverage of datum δ changes over time, snapshots of which compose the *coverage dynamics* $\{C(\delta, t)\}_{t \in \mathcal{T}} \subset A^{\mathcal{T}}$ that describes the *whereabouts* of datum δ during its dissemination process. Based on this data coverage model, we formulate the 'where' question into three sub-questions, with respect to $\{C(\delta, t)\}_{t \in \mathcal{T}}$:

- *Quantification*: How to describe/calculate the data coverage $C(\delta, t)$ at time instant t?
- *Evolution*: What is the impact of user mobility and data forwarding actions on the change of data coverage over a time step, *e.g.*, from $C(\delta, t)$ to $C(\delta, t+1)$?
- *Prediction*: OBServing the dynamics $\{C(\delta, t)\}_{t=1}^{T}$ for T time steps, can we predict its evolution in the future, that is, estimating data coverage $C(\delta, s)$ for $s > T$?

Among these, the first question calls for a quantification of coverage dynamics, which is a premise to address the rest two problems. Therefore in the next section, we introduce our analysis framework that formulates the dynamics into a structured set of numeric random processes, before addressing the evolution and prediction problems.

3.3 Representing Coverage Dynamics with Graph Signals

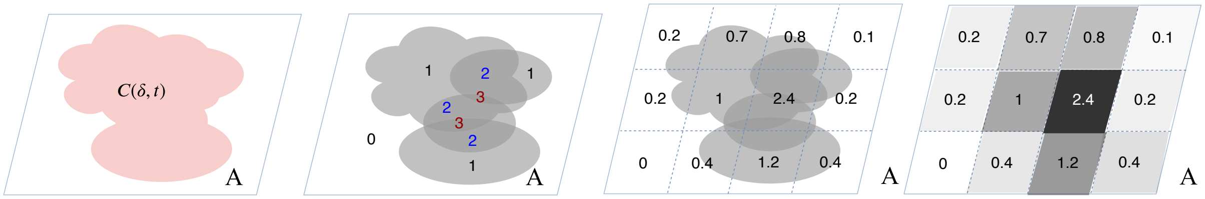
By definition in the last section, data coverage of datum δ is the union of individual coverage, $\cup_{e \in E} C_e(\delta, t)$. Though intuitive, this entity-centric definition with the union operator does not naturally provide any quantitative measure on coverage. In other words, we can not calculate, or compare data coverage instances. For example, we can not compare coverage of datum δ at different time instants, $C(\delta, t)$ and $C(\delta, t+1)$, or compare data coverage of different data blocks at the same instant, *e.g.*, coverage $C(\epsilon, t)$ and $C(\delta, t)$ of ϵ and δ, respectively.

Therefore, to answer the *quantification* question, we make coverage dynamics tractable by a numeric measure of data coverage, and then discuss entity-level state transitions, and system-level governing equations, as the analysis framework of data coverage dynamics.

3.3.1 A Location-centric Measure: Data-strength

The cornerstone of this framework is a metric to represent data coverage $C(\delta, t)$ numerically, through which the data coverage (a set of infinite location points) of an irregular shape, *e.g.*, $C(\delta, t)$ (as shown in Figure 3.3a), can be compared with its predecessor $C(\delta, t-1)$ for further analysis. Ideally, this metric should reflect the *strength* of datum δ's coverage at time t. For example in Figure 3.3b, darkness of the shade indicates the number of datum copies that can be obtained at different locations: the darker the shade, the more probable that a copy of datum δ can be obtained during time step $t + 1$.

To do so, we change from an entity-centric perspective, to a location-centric perspective. First, partition region A into a grid of N non-overlapping sub-regions (later referred to as *cells*)



(a) Data coverage at time t, $C(\delta, t)$, is of an irregular shape.

(b) Strength of the coverage in terms of number of datum copies.

(c) Partition of region A into a grid of $N = 16$ square cells.

(d) Data-strength $\mathbf{s}(\delta, t) \in \mathbb{R}_0^N$ is a vector of N real numbers.

Figure 3.3 Data-strength $\mathbf{s}(\delta, t)$ describes data coverage $C(\delta, t)$ at the space granularity of cells.

$\{A_1, A_2, \cdots, A_N\}$, as shown in Figure 3.3c. Then for each cell A_n, measure the **data-strength** as

$$s^n(\delta, t) := \sum_{e \in E} \frac{|A_n \cap C_e(\delta, t)|}{|A_n|}, \tag{3.3}$$

where $|\cdot|$ denotes the area of a region.

The the assembled vector of data strength, $\mathbf{s}(\delta, t) = [s^1, s^2, \cdots, s^N](\delta, t)$, can fully describe data coverage at time t, in the sense that:

- The location information (*where*) is encoded in the superscripts of vector $\mathbf{s}(\delta, t)$, such that changes in the space domain are transformed into variation of numeric values;
- Given vector $\mathbf{s}(\delta, t)$, data coverage $C(\delta, t)$ can be approximated by the the union of cells in set $\{A_n \mid s^n(\delta, t) > \alpha\}$, where $\alpha \geq 0$ is a predetermined threshold;
- The data-strength metric measures the expected number of copies of datum δ an entity can gather if it is located in a cell. So data-strength of a cell can be viewed as the intensity of data coverage, at the space granularity of cells.

Remark 3.1. *In some cases, it may not be necessary, or possible to count the exact number of data carriers in a cell, especially due to privacy concerns. For example, a carrying entity, who agrees to share data (participate in the dissemination) anonymously, may be reluctant to reveal its carrying status to others. For these cases, data strength can be measured from the perspective of recipients, that is, the number of datum δ an entity can receive in a cell during a time step. In other words, the system operator can employ probe entities to measure data strength.*

Remark 3.2. *In theory, any form of cell is acceptable. In fact, cells don't even have to be mutually exclusive, because as long as set $\mathcal{A} = \{A_1, A_2, \cdots, A_N\}$ jointly cover region A, vector $\mathbf{s}(\delta, t)$ will be a numerical description of data coverage $C(\delta, t)$ over the whole region A. From the data owner's perspective, it is also sensible to include all the sub-regions of interest as the collection $\mathcal{A}$, e.g., entrances, elevators and stairways of a mall in the mobile ad application. We adopt the partition of square cells for its simplicity in denotation and simulation.*

When the area of a cell is larger than, or comparable to, the area of individual coverage, data-strength of cell A_n can be further simplified[1] to

$$s^n(\delta, t) := \sum_{e \in E} \delta_e(t) \mathbb{1}_{A_n}(p_e(t)) . \tag{3.4}$$

In this way, data coverage, a time-varying subset of A, is transformed into a numeric data-strength vector, whose value can be calculated with attributes $\{\delta_\cdot(t), p_\cdot(t)\}_{e \in E}$ of all the entities. Next, we discuss the short-term evolution of the coverage dynamics, that is, how data strength $\mathbf{s}(\delta, t)$ over the region A changes over time, from the entity-level and system level, respectively.

3.3.2 State Transitions of a Single Entity

As can be seen in the state transition diagram (grey box in Figure 3.1) of an entity, individual state changes are driven by the decision an entity makes, which is impossible for a large system to determine at an individual level. Therefore, we characterize their preference on the data dissemination process at a system-level, with participating rate β and withdrawing rate γ.

(1) *Participating rate* $\beta \in [0, 1]$ is the probability that an arbitrary entity $e \in E$ chooses to *participate* during time step t, given that e has received one copy of datum δ during time step $t-1$. In other words, for an entity $e|_{\delta_e(t-1)=0,\ p_e(t-1) \in A_n}$ located in cell A_n at time $t-1$, the probability that its state transits from 0 to 1 during time step t can be obtained as

$$\mathbb{P}\Big(\delta_e(t) = 1 \;\Big|\; \{\delta_e(t-1) = 0, p_e(t-1) \in A_n\}\Big) \simeq 1 - (1-\beta)^{f(\mathbf{s}(\delta,t)) + \sum_{x \in E} \delta_x(t-1) B_{x,e}}, \tag{3.5}$$

where $f(\mathbf{s}(\delta, t)) = s^n(\delta, t-1) + c \sum_{j \in N(n)} s^j(\delta, t-1)$, in which c is a coefficient and $N(n)$ denotes the indices of cells that are adjacent to cell A_n; and quantity $B_{x,e} = 1$ if and only if entity e is a stationary entity, and is connected to another stationary entity x via a wired link. In fact, the exponent in Eq. (3.5) equals to the number of datum copies that entity e expects to receive over wireless links during time step $t-1$, which includes datum copies from co-locating entities (inside cell A_n) and those from adjacent cells (only a fraction c of datum copies sent from entities that are close to cell A_i can be heard by e). For mobile entities, the last term in the exponent of Eq. (3.5) equals to 0.

For the clarity of description, we refer to the heterogeneous scenario as Case-2, which includes both wired and wireless connections as described in Eq. (3.5), and refer to the simpler scenario with merely wireless connections as Case-1, throughout this chapter.

(2) *Withdrawing rate* $\gamma \in [0, 1]$ is the ratio of participating entities that choose to stop disseminating data in the next time step, or equivalently, the probability that an entity $e|_{\delta_e(t-1)=1}$

[1] Entities in cell A_n can also obtain copies of δ from carrying entities in adjacent cells, but this part of data-strength is 'lost' due to partition, which is negligible when cells are large (compared to individual coverage of single entities). The impact of cross-cell data forwarding is considered in Section 3.3

takes a *withdraw* action (resulting in a state transition from 1 to 0), that is,

$$\mathbb{P}\big(\delta_e(t) = 0 \mid \delta_e(t-1) = 1\big) = \gamma, \ \forall t \in \mathcal{T}. \tag{3.6}$$

Remark 3.3. *Note that the probability of a participate action increases with the total number datum copies an entity receives, as stated in Eq.* (3.5)*, while the probability to withdraw is independent with actions of other entities, which can be seen from Eq.* (3.6)*. The reason behind the increasing willingness to participate over the number of received data copies is that, more received copies indicate there are more participants in the system, which further implies a possibly high incentive offered by the data owner, hence the larger chance of participating. On the other hand, an entity already knows the incentive when it makes the decision on withdrawal, so its decision to withdraw will not be affected by others' actions. Instead, an entity could withdraw from the dissemination process of datum δ due to lack of storage space, lost of network connectivity, or simply out of its own will, which are captured by probability γ as a whole.*

As system-level indicators of an entity's willingness to participate, rate β and γ can be changed by adjusting the incentives offered to participants. With these rates, we link the individual states $\delta_e(t)$ to the data-strength $\mathbf{s}(\delta, t-1)$ of a single cell, via Eq. (3.5) and (3.6). Next, we zoom out further to analyze the dynamics' evolution on the system level.

3.3.3 Evolution of the Dynamics via Data-strength Vector $\mathbf{s}_t$

The key to understand the coverage dynamics $\{C(\delta, t)\}_{t\in\mathcal{T}}$ is knowing how it evolves in one time step, *i.e.*, from $C(\delta, t-1)$ to $C(\delta, t)$, or equivalently under the data-strength measure, from vector $\mathbf{s}(\delta, t-1)$ to $\mathbf{s}(\delta, t)$. For the ease of notation, we omit δ, and write $\mathbf{s}(\delta, t)$ as $\mathbf{s}_t = [s_t^1, s_t^2, \cdots, s_t^N]^\intercal$, as we are only interested in the dissemination process of datum δ.

Consider cell $A_n \in \mathcal{A}$. The change of data-strength in this cell, $(\Delta \mathbf{s}_t)^n = s_t^n - s_{t-1}^n$, is induced by two impetuses, that is, movements of entities (across cells), and data dissemination (inside cell A_n and across neighboring cells), which we discuss separately.

(1) *User mobility component.* Let $p_t^n := \sum_{e\in E} \mathbb{1}_{A_n} p_e(t)$ denote the number of entities inside cell A_n at time t. We write its vector form as $\mathbf{p}_t = [p_t^1, p_t^2, \cdots, p_t^N]$. Let $m_{i,j}^t$ denote the number of entities that move out of cell A_i to a different cell A_j during time step t, *i.e.*, $m_{i,j}^t = |\{e \in E | p_e(t-1) \times p_e(t) \in A_i \times A_j\}|$. For cell index $i \neq j$, define the i,j-th element of a $N \times N$ matrix $\mathbf{W}_t$, the user mobility matrix, as

$$W_{i,j}^t := \begin{cases} \frac{m_{i,j}^t}{p_t^n}, & \text{if } i \neq j \text{ and } p_t^n > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{3.7}$$

Matrix $\mathbf{W}_t$ records the ratio of entities that move across cell boarders. During time step t, among the $W_{n,k}^t p_t^n$ entities that moved from cell A_n to cell A_k, on average, there are $(1-\gamma) W_{n,k}^t s_t^n$ data carriers, which contributes to the decrease in data-strength s_t^n, and the increase of s_t^k.

(2) *Data dissemination component.* Assuming entities are 'well-mixed' in each cell[2], we apply MFA at the cell level to derive the one-step evolution equation. We start from the simpler Case-1 with merely wireless connections. For cell A_n, on average, the number of new participants grows at rate $\beta f(\mathbf{s}(\delta, t))$ entities per time step, and existing participants withdraw at rate γs_{t-1}^n entities per time step.

With the entity-level state transition analysis and MFA, the average evolving trend of the coverage dynamics can be described with $\mathbf{W}_t$, $\mathbf{s}_t$ and $\mathbf{p}_t$. For every cell A_n in a Case-1 scenario, the number of entities p_t^n and data-strength s_t^n satisfy

$$p_t^n = p_{t-1}^n + \sum_{j=1}^{N} W_{j,n}^t p_{t-1}^j - \sum_{k=1}^{N} W_{n,k}^t p_{t-1}^n, \tag{3.8}$$

$$s_t^n = (1-\gamma)\left(s_{t-1}^n + \sum_{j=1}^{N} W_{j,n}^t s_{t-1}^j - \sum_{k=1}^{N} W_{n,k}^t s_{t-1}^n\right) + (p_{t-1}^n - s_{t-1}^n)\beta\left(s_{t-1}^n + c\sum_{m\in N(n)} s_{t-1}^m\right). \tag{3.9}$$

A clear observation from Eq. (3.9) is that the signal value of a cell at time t, is the summation of a scaled version of its previous value, and the influence of its *neighboring* cells, which are captured by the user mobility matrix $\mathbf{W}_t$ and $N(n) \subset \mathcal{A}$. For the Case-2 scenario, the number of entities in each cell p_t^n still satisfies Eq. (3.8), while the data strength may increase due to data transmission along wired connections, which are time-invariant, and accessible before the data dissemination process. Denote $\mathrm{AP}_n := \{e \in E \mid p_0^e = p_1^e = \cdots = p_t^e \in A_n\}$ as the set of stationary entities in cell A_n, and $L_{n,j} := \sum_{e,x\in N^2} \mathbb{1}_{\mathrm{AP}_n}(e) B_{ex} \mathbb{1}_{\mathrm{AP}_j}(x)$ as the number of wired connections between cell A_n and A_j. Recall B_{ex} is defined as 1 when entity e and x are connected permanently by a wired link, in Eq. (3.5). Then data strength s_t^n in the Case-2 scenario has one more term than Eq. (3.9), *i.e.*,

$$\begin{aligned} s_t^n = (1-\gamma)&\left(s_{t-1}^n + \sum_{j=1}^{N} W_{j,n}^t s_{t-1}^j - \sum_{k=1}^{N} W_{n,k}^t s_{t-1}^n\right) \\ &+ (p_{t-1}^n - s_{t-1}^n)\beta\left(s_{t-1}^n + c\sum_{m\in N(n)} s_{t-1}^m\right) + \frac{p_{t-1}^n - s_{t-1}^n}{p_{t-1}^n}\beta\sum_{j=1}^{N} L_{n,j}\frac{s_{t-1}^j}{p_{t-1}^j}. \end{aligned} \tag{3.10}$$

From Eq. (3.10), we can see that, the time-invariant matrix $\mathbf{L}_{N\times N}$ adds another layer of interconnection between cells (in addition to the user mobility matrix $\mathbf{W}_t$), which is independent of geographical adjacency (as apposed to matrix $\mathbf{W}_t$). Together, Eq. (3.8), (3.9), and (3.10) sketch the average evolution trend of the data-strength with the following implications.

(1) Mobility drives the coverage dynamics. The change in data-strength during time step

[2] 'Well-mixed' means each entity has an equal chance of contact (transmit/receive datum) with any other entities in the same cell. We set the cell size to be comparable to individual coverage (in terms of area), so that every entity in a cell is almost under the individual coverage of each other.

t, $\Delta \mathbf{s}_t = \mathbf{s}_t - \mathbf{s}_{t-1}$, relies heavily on user mobility (captured by matrix $\mathbf{W}_t$ in Eq. (3.8) and (3.9)), which is to say, the change in the data strength of each cell can be written as a function of $\mathbf{W}_t$. This is especially true when the participating rate β (indicator of data forwarding) is small or the withdrawing rate γ is small, as both the last term in Eq. (3.9) and the extra term in Eq. (3.10) also tend to be small, compared to the first term in Eq. (3.9). In addition, term $\sum_{m \in N(n)} s_{t-1}^m$ in Eq. (3.9) can be re-written as a function of $\mathbf{W}_t$ and $\mathbf{s}_{t-1}$, because cross-cell movements (in one time step) are only possible, if the two cells are close to each other.

(2) Data-strength $\mathbf{s}_t$ is a **graph signal**. By nature, any data-strength vector $\mathbf{s}_t$ is essentially a vector of positive real numbers indexed by cells, *i.e.*, a *signal* defined on a graph with vertex set $\mathcal{A}$. Given an initial state $\mathbf{s}_0$, the signal value s_t^n of cell A_n can be estimated with the data-strength in the last time step, *i.e.*, $\mathbf{s}_{t-1}$ and matrix $\mathbf{W}_t$, as shown in Eq. (3.9). In fact, only a few neighboring cells (determined by elements of $\mathbf{W}_t$) are relevant to the change of data-strength during time step t, because correlation between signal values s_t^n and s_{t-1}^m decreases as the distance between cell A_n and A_m increases, which is true for both Eq. (3.8) and Eq. (3.9). Therefore, matrix $\mathbf{W}_t$ is closely related to the underlying structure of signal $\mathbf{s}_t$, and can be viewed as the adjacency matrix of the graph. Formulating $\mathbf{s}_t$ as a graph signal allows us to explore the rich property of coverage dynamics in both the space domain and the time domain, as a preparation to which, we briefly introduce the basics of GSP for self-containment reasons.

3.3.4 Preliminaries on Graph Signal Processing (GSP)

graph signal processing (GSP) is an emerging tool to describe, analyze, and recover high-dimensional data. On this avenue, Sandryhaila and Moura extended digital signal processing (DSP) concepts to static signals index by graphs, and formally defined the graph shift operator, graph fourier transform and graph filters based on adjacency matrices [105]. Similar concepts are also defined based on the graph Laplacian [108], which has nice properties such as positive semi-definite, to make GSP more commutable with DSP. Then taking the evolution of graph signals into consideration, Grassi *et.al.* constructed a time-vertex signal processing framework, which formally defined time-vertex process, joint time-vertex Fourier transform (JFT) [75], and filter banks [45]. Based on this framework, prediction of stationary graph signal with frequency domain analysis is made possible in [75], and proved to have a lower complexity. Recently, Marques *et.al.* generalized the definition of weak stationarity for random graph signals, and proposed a spectral estimation mechanism to predict real-world graph signals [79].

Next, we briefly introduce the definition and terminologies in GSP that are useful in this chapter. Interested users are readers are directed to [108, 105, 45, 75, 79] for details.

3.3.4.1 Graph Fourier Transform (GFT) and Spectrum of a Graph Signal

Consider graph signal $\mathbf{s}_t \in \mathbb{R}^N$ on an undericted and connected graph $\mathcal{G}(\mathcal{A}, \mathbf{W})$, in which $\mathcal{A}$ is the vertex set, and $\mathbf{W}$ is the weighted adjacency matrix of $\mathcal{G}$. Let $\mathbf{D}$ be the generalized degree matrix of $\mathcal{G}$, that is, a diagonal matrix with diagonal elements $\mathbf{D}_{i,i} = \sum_{j \in \mathcal{A}} w_{i,j}$. Then the **Laplacian** of graph $\mathcal{G}$ is defined as $\mathcal{L}_{\mathcal{G}} := \mathbf{D} - \mathbf{W}$. Since $\mathcal{G}$ is undericted, $\mathcal{L}_{\mathcal{G}}$ is real-valued, symmetric and positive semi-definite, as a result of which, $\mathcal{L}_{\mathcal{G}}$ has N real, non-negative eigenvalues $\{\lambda_n\}_{n=0,1,\ldots,N-1}$.

Without loss of generality, let the eigenvalues $\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{N-1}$ be non-decreasingly ordered, and denote $\Lambda = diag(\lambda_0, \lambda_1, \cdots, \lambda_N)$. Then we always have $\lambda_0 = 0$, and each eigenvalue λ_n corresponds to an eigenvector $\mathbf{v}_n$ (column vector). Let $\mathbf{U}_{\mathcal{G}} = [\mathbf{v}_0, \mathbf{v}_1, \cdots, \mathbf{v}_{N-1}]$ be the matrix form of all the eigenvectors of $\mathcal{L}_{\mathcal{G}}$. Then graph Laplacian $\mathcal{L}_{\mathcal{G}} = \mathbf{U}_{\mathcal{G}} \Lambda \mathbf{U}_{\mathcal{G}}^*$. The **graph Fourier transform (GFT)** of signal $\mathbf{s}_t$ is defined as the product of $\mathbf{U}_{\mathcal{G}}$ and the graph signal $\mathbf{s}_t$, that is,

$$\widehat{\mathbf{s}_t} = \mathrm{GFT}\{\mathbf{s}_t\} := \mathbf{U}_{\mathcal{G}}^* \mathbf{s}_t, \tag{3.11}$$

where $\mathbf{U}_{\mathcal{G}}^*$ is the conjugate transpose of $\mathbf{U}_{\mathcal{G}}$, and $\widehat{\mathbf{s}}_{t N\times 1} = [\mathbf{s}_t(\lambda_0), \mathbf{s}_t(\lambda_1), \cdots, \mathbf{s}_t(\lambda_{N-1})]^\intercal$ is the GFT spectrum of $\mathbf{s}_t$. Like the spectrum of a signal in the 1-dim time domain, GFT spectrum $\widehat{\mathbf{s}_t}$ of graph signal $\mathbf{s}_t$ captures the *variation* of the signal in the vertex domain ($\mathcal{A}$).

3.3.4.2 Time-Vertex Process and Stationarity

Temporal evolution of a graph signal composes a time-vertex process $\mathbf{S}_{N\times T} = [\mathbf{s}_1;\ \mathbf{s}_2;\ \cdots;\ \mathbf{s}_T]$.

Along the time dimension, every row vector of $\mathbf{S}$, can be analyzed through discrete Fourier transform (DFT). The DFT of process $\mathbf{S}$ is defined as $\mathrm{DFT}\{\mathbf{S}\} := \mathbf{S}\overline{\mathbf{U}_T}$, where $\mathbf{U}_T$ is the T-point normalized DFT matrix. Its (t, s)-element is $\mathbf{U}_T(t, s) = \frac{e^{-j\omega_s t}}{\sqrt{T}}$, and $\omega_s = \frac{2\pi(s-1)}{T}$.

Grassi *et.al.* defined the joint Fourier transform (JFT) [45] as $\mathrm{JFT}\{\mathbf{S}\} := \mathbf{U}_{\mathcal{G}}^* \mathbf{S} \overline{\mathbf{U}_T}$. The time-vertex process $\mathbf{S}$ is joint wide-sense stationary (JWSS), if it is wide-sense stationary in both the vertex domain and the time domain [75, 47]. Or formally as stated in [75], process $\mathbf{S}$ is JWSS with respect to graph $\mathcal{G}$, if an only if (i) $\mathcal{L}_J \mathbb{E}(\mathrm{vec}(\mathbf{S})) = \mathbf{0}$, where the joint Laplacian $\mathcal{L}_J = \mathcal{L}_{\mathcal{G}} \times \mathcal{L}_T$; and (ii) its covariance matrix $\Sigma_{\mathrm{vec}(\mathbf{S})}$ is diagonalizable by the joint basis $\mathbf{U}_{\mathcal{G}} \otimes \mathbf{U}_T$.

Recall in Section 3.3.3, we find that user mobility drives the evolution of data coverage dynamics. In regard to user mobility and data dissemination, Grossglauser and Tse proved that mobility increases per-user throughput in ad hoc wireless networks [46], when mobile nodes function as relays. Naturally, we anticipate data coverage to be boosted by an increase in mobility (speed), because: i) In both Case-1 and Case-2 scenario, entities function as relay, passing the datum to others. ii) Increased mobility leads to an increase of probability that a relay can meet with a destination (in this case, any entity), as well as pair-wise throughput [46], which indicates that more data transmissions between entities can happen simultaneously. But an open question is, *how much* does mobility (captured by $\mathbf{W}_t$) and data forwarding (captured by β) contribute to the change of data-strength $\Delta\mathbf{s}_t$?

3.4 Information from a Snapshot

Knowing quantitatively how much entity mobility and data forwarding contribute to the dynamic evolution of data coverage can significantly benefit data disseminators, who expect to boost data coverage effectively. For example, which works better for an ad distributor: raise incentive to increase the participating probability β, or employ faster-moving entities as ad distributors? This is especially meaningful to the Case-1 scenario, in which wired connections are not available for data dissemination processes.

Therefore, in this section, we discuss the impact of user mobility in the Case-1 scenario. To be more specific, we examine a snapshot of the coverage dynamics, *i.e.*, the difference graph signal $\Delta\mathbf{s}_t = \mathbf{s}_t - \mathbf{s}_{t-1}$, with respect to user mobility $\mathbf{W}_t$. A simple homogeneous case with a synthetic mobility model is considered in this section, to better uncover the impetuses.

3.4.1 A Simple Homogeneous Scenario

Suppose the entire network is composed of mobile entities that i) have the same transmission range r_c, and ii) move according to a speed-constrained mobility model detailed as follows. During the 'move' stage in every time step, each entity $e \in E$ chooses a direction uniformly at random from $[0, 2\pi)$, along which e moves at a speed randomly chosen from $[0, v_{max}]$. We follow the convention in [66] and assume that, an entity crossing the boundary of region A on one side will emerge from the opposite side. In addition, considering that every entity withdraws independently, we set the withdrawal rate $\gamma = 0$ in this section to focus on the impact of maximum speed v_{max}, participating rate β, and communication range r_c.

3.4.2 Impact of Mobility

Recall data-strength vector $\mathbf{s}_t$ (similarly the difference signal $\Delta\mathbf{s}_t$) is a graph signal defined on vertex set $\mathcal{A}$ of some graph $\mathcal{G}$. Also, we observed the important role of user mobility matrix $\mathbf{W}_t$ in the dynamical change of data strength, in Eq's. (3.8) to (3.10), so we propose to define the weighted adjacency matrix $\widetilde{\mathbf{W}_t}$ of graph $\mathcal{G}$ as a function of $\mathbf{W}_t$, which is fairly accessible (due to prevalent GPS devices [103], IoT sensors [104], and location-based services [110]).

3.4.2.1 Weighted Adjacency Matrix $\widetilde{\mathbf{W}_t}$ of $\mathcal{G}$

The evolution rule in Eq. (3.9) is formulated based on the instantaneous user mobility $\mathbf{W}_t$, but it is not suitable to directly assign $\mathbf{W}_t$ as the adjacency matrix of $\mathcal{G}$ for real systems, because of the time gap between data transmission (several ms to seconds) and positioning (*e.g.*, GPS logging, seconds to minutes). To bridge this gap, we observe the system every $K \in \mathbb{N}^+$ time steps, and obtain a snapshot $\Delta\mathbf{s}_t^K = \mathbf{s}_t - \mathbf{s}_{(t-K+1)}$ for every observation. Accordingly, we employ the symmetric accumulated version $\widetilde{\mathbf{W}_t}$ as the weighted adjacency matrix of graph $\mathcal{G}$, defined as the number of cell crossings during time interval of length K.

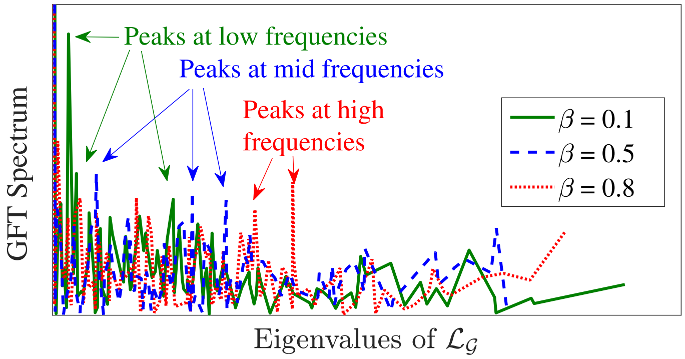


Figure 3.4 GFT spectrum of three difference signals $\Delta\mathbf{s}_t$, generated with different β values, illustrate the variation of $\Delta\mathbf{s}_t$ over mobility $\widetilde{\mathbf{W}}_t$: the higher the participating probability β, the more dominant the high frequency components, and the less impact by user mobility.

Remark 3.4. *We use the number of crossing, instead of ratio of crossing in Eq.* (3.7)*, due to two reasons: i) it is easier to obtain offline for a period of time, and ii) the number of cell crossings also captures correlation between adjacent cells (*$N(n)$ *in Eq.* (3.9) *and* (3.10)*), because the more cell crossing, the more entities gathering at the cell edge, and hence a higher probability of data forwarding across cell boarders. In this way, we are able to reduce the number of variables in Eq.* (3.9) *and* (3.10)*, and write* $\mathbf{s}_t = \left[(1-\gamma+\beta)^K\mathbf{I} + f(t)\widetilde{\mathbf{W}}_t\right]\mathbf{s}_{t-K+1} + \vec{\psi}_t$ *instead, where* $f(t)$ *is a coefficient function by* β *and* γ*, and* $\vec{\psi}_t$ *is the error term introduced by the approximation.*

To validate the graph formation based on $\widetilde{\mathbf{W}}_t$, and the speculation that the change in data-strength, $\Delta\mathbf{s}_t^K$, is a graph signal on $\mathcal{G}$, we first examine three snapshot instances (with the same user mobility and initial state, so graph $\mathcal{G}$ and the weight matrix $\widetilde{\mathbf{W}}_t$ are exactly the same). As can be seen from the GFT (defined in [108, 105]) spectrum in Figure 3.4, high participating rate β will induce peaks at high frequencies, that is, larger products $\widehat{\Delta\mathbf{s}}(\lambda_n) =< \Delta\mathbf{s}, \mathbf{v}_n >$ at larger eigenvalues λ_n of the graph Laplacian $\mathcal{L}_{\mathcal{G}}$ of $\mathcal{G}$. This conceptually validates our observation of $\Delta\mathbf{s}_t^K$ being a graph signal on $\mathcal{G}$, which is defined by matrix $\widetilde{\mathbf{W}}_t$. However, spectrum $\widehat{\Delta\mathbf{s}}$ of signal $\Delta\mathbf{s}_t^K$ does not intuitively or quantitatively reflect the impact of user mobility. Therefore, we define a new metric to quantify the impact.

3.4.2.2 Mobility Dependence Index (MDI)

With user mobility quantitatively represented by $\widetilde{\mathbf{W}}_t$, the level of *inconsistency* of difference signal $\Delta\mathbf{s}_t$ with respective to $\mathcal{G}$ identifies the impetus that is not from mobility, *i.e.*, from data forwarding, as illustrated in Figure 3.4. To quantify the impact of mobility (captured by $\widetilde{\mathbf{W}}_t$), we define a new metric for every interval $[t, t+K-1]$.

Definition 3.2. *The* ***mobility dependency index*** *of a snapshot* $\Delta\mathbf{s}_t^K$ *with respective to* $\widetilde{\mathbf{W}}_t$ *is defined as*

$$\mathrm{MDI}_{\widetilde{\mathbf{W}}_t}(\Delta\mathbf{s}_t^K) := \frac{1}{\|\Delta\mathbf{s}_t^K - \frac{1}{\lambda_{max}}\widetilde{\mathbf{W}}_t\Delta\mathbf{s}_t^K\|_1}, \tag{3.12}$$

where λ_{max} *is the maximum eigenvalue of* $\widetilde{\mathbf{W}}_t$*.*

In this definition, the denominator is the *total variation* $\text{TV}_{\mathcal{G}}$ of the difference signal $\Delta\mathbf{s}_t^K$. Intuitively, $\text{TV}_{\mathcal{G}}$ quantifies the conformity of $\Delta\mathbf{s}_t^K$ with respect to entity movements captured by weight matrix $\widetilde{\mathbf{W}}_t$. Therefore, the higher the MDI (*i.e.*, lower $\text{TV}_{\mathcal{G}}$), the more mobility contributes to the evolution of coverage dynamics. When the difference signal $\Delta\mathbf{s}_t^K$ is very smooth (with respect to graph $\mathcal{G}$), the change in data-strength is aligned with weight matrix $\widetilde{\mathbf{W}}_t$, indicating a heavy influence of mobility. In contrast, any increment in signal value attributed to data forwarding will induce an increment in $\text{TV}_{\mathcal{G}}$, and hence a decrease in the MDI value.

3.4.2.3 Simulation Configuration

To observe the impact of user mobility via the MDI metric, we simulate a dissemination process in the wireless connection only, Case-1 scenario, whose configurations shown in Table 3.1.

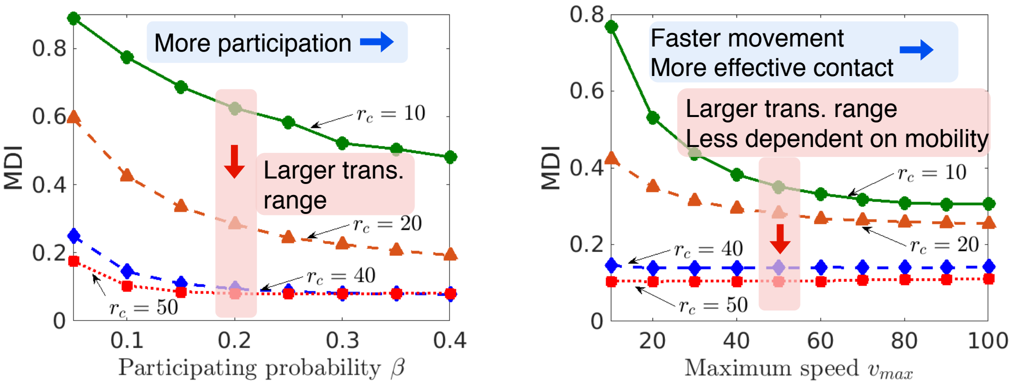
Table 3.1 Simulation settings for Section 3.4.

Para.	Description	Value	Para.	Description	Value
A	region of interest	$[0, 1000]^2$	$\|E\|$	# of entities	1000
$\|A_i\|$	area of a cell	100×100	N	# of cells in A	100
r_c	communication range	[10, 50]	v_{max}	max. speed	[10, 100]
N/A	# of seeds	50	N/A	# of runs	100
β	participating prob.	[0.05, 0.4]	γ	withdrawing prob.	0
Δ	time step	1 s	K	observation interval	5 s

3.4.2.4 Observations

Figure 3.5 shows the MDI value of the difference signal $\Delta\mathbf{s}_t^K|_{K=5}$ generated with the same initial condition $\mathbf{s}_t$, but with different participating rate β, maximum speed v_{max} and transmission range r_c. We highlight the following observations.

(1) *Trend of MDI*: It is intuitive that MDI decreases when transmission range r_c increases (Figure 3.5a and b), and also decreases when the participating probability β increases (Figure 3.5a), because increment in either r_c or β will result in more non-carrying entities ($\delta_e(t) = 0$) to receive a copy of δ and participate in the dissemination. Nonetheless, it is rather interesting to observe in Figure 3.5b, that when the maximum speed v_{max} increases, MDI decreases dramatically for small transmission ranges. The reasons behind this plunge are: i) the increment in data-strength largely relies on new effective contacts, that is, data forwarding between entity-pairs, which are more prevalent due to the larger movement range ($K \cdot v_{max}$); and ii) elements in user mobility matrix $\widetilde{\mathbf{W}}_t$ can not capture geographical adjacency of cells any more, as movements between non-adjacent cells are now possible due to the larger movement range.



(a) MDI vs. β ($v_{max} = 10$) **(b)** MDI vs. v_{max} ($\beta = 0.1$)

Figure 3.5 Bounded impact of both mobility and data forwarding on the evolution of data coverage dynamics (MDI values in both figures are scaled to be more visible).

(2) *Decreasing, but bounded impact*: observe that MDI flats out as β and v_{max} further increases, indicating a *threshold behavior* of the coverage dynamics with respect to both impetuses: the change of data coverage can not always be accelerated by increasing β or v_{max}. In fact, above a certain threshold, *e.g.*, $\beta = 0.2$ at communication range $r_c = 40$, further increasing β will not decrease the MDI in Figure 3.5a, so β has bounded impact on the coverage change. Similar effect is also seen for the maximum speed v_{max}.

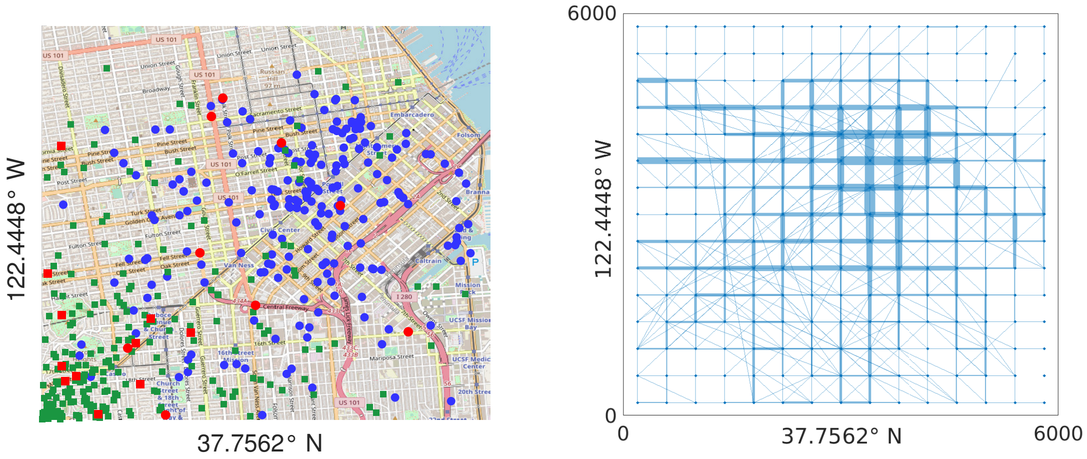
(3) *Prediction based on* $\widetilde{\mathbf{W}}_t$: observe the high MDI when maximum speed v_{max} (and preferably the participating rate β as well) of entities is relatively low, which is also captured by Eq. (3.9) and (3.10). This implication motivates us to consider predicting the data coverage based on the weight matrix $\widetilde{\mathbf{W}}_t$, which is quite accessible in a real-world edge network.

3.5 Prediction of Data Coverage: Where Data Are

As a preliminary application of the proposed coverage dynamics model, and observed properties on its evolution, we examine a realistic heterogeneous edge network, and design a framework to answer the *prediction* question, *i.e.*, *where data will be.*

3.5.1 A Real-world Heterogeneous Edge Network

It would be ideal to experiment with data coverage collected from operating networks. However, traces with data forwarding logs are usually not available due to privacy concerns, while mobility traces are abundant. Therefore, we first generate a half-synthetic coverage dynamics trace by simulating a data dissemination process with real-world user mobility traces.



(a) Initial state at $t = 0$ s.

(b) Graph $\mathcal{G}$ (edge weight shown by width).

Figure 3.6 A data dissemination process by taxis (blue dots) and WiFi hotspots (green squares). Initial datum carriers (at time 0) are marked in red.

3.5.1.1 Entities and Mobility

There are two types of entities in this network: 1072 taxis[3] that move according to their GPS logs in the `cabspotting` dataset [93], and 231 WiFi APs that are positioned according to the urban hotspot model proposed in [106]. Moving speed of taxis satisfies a Weibull(5.88, 1.60) distribution with mean 5.88 m/s, which is quite low (compared to Figure 3.5 (b)), so MDI is high. In the Case-2 scenario, each pair of APs is connected by a wired link with probability 0.5.

3.5.1.2 Region A and Initial Conditions

We consider a 6 Km by 6 Km region A in downtown San Francisco, which is divided into a (15 × 15) grid of square cells, as shown in Figure 3.6a. At time $t = 0$, 50 entities (seeds) are randomly selected to receive the datum. Note that taxis may move out of region A. In fact, the GPS records in the `cabspotting` dataset expands far beyond region A, as a result of which data dissemination can happen outside region A as two taxis meet. These events are also considered and simulated in our experiment (but not included in the signal value collection), to better mimic a real-world system.

3.5.1.3 Weight Matrix W

For the ease of implementation, we consider the average weight matrix $\mathbf{W}$ as follows:

$$\mathbf{W} := \frac{1}{\lceil \frac{T}{K} \rceil} \sum_{j=1}^{\lceil \frac{T}{K} \rceil} \widetilde{\mathbf{W}}_{jK+1} \tag{3.13}$$

[3] The `cabspotting` dataset contains 536 taxis. Considering they move out of region A easily, we replicate the logs in reverse time, such that the total number of taxi is 1072. In addition, the GPS logs are also truncated to 12000 s to align with the shortest individual log.

The reason to use the long-term average $\mathbf{W}$, instead of $\widetilde{\mathbf{W}_t}$, is that it is easier to obtain (collect and calculate offline) one copy of $\mathbf{W}$, and use it for all analysis, as the moving patterns of taxis are quite stationary over a time span of 4 hours (determined by the `cabspotting` dataset). The graph $\mathcal{G}(\mathcal{A}, \mathbf{W})$ constructed based on weighted adjacency matrix $\mathbf{W}$ is shown in Figure 3.6b, where width of an edge (blue line segments) is proportionate to its associated weight in $\mathbf{W}$. As can be seen from Figure 3.6b, the correlation between directly adjacent cells are well captured by the mobility pattern (thicker horizontal and vertical lines), while the diagonal or off-diagonal adjacency is not prominent (thinner leaning lines). The reason behind this is that $\mathbf{W}$ is constructed based on GPS logs of taxis, who travel along road segments, which are usually horizontal or vertical, as can be seen from Figure 3.6a. It is especially suitable in this case[4], because the we consider a grid of square cells, which resemble street blocks.

3.5.2 Coverage Dynamics as a Time-vertex Process

On graph $\mathcal{G}(\mathcal{A}, \mathbf{W})$, observations of the coverage dynamics, *i.e.*, signal $\mathbf{s}_t$ constitute a *time-vertex* process [45] over time span $[1, T]$. We can write this process $\{\mathbf{s}_t\}_{t\in[1,T]}$ in its matrix form $\mathbf{S}_{N\times T} = [\mathbf{s}_1;\ \mathbf{s}_2;\ \cdots;\ \mathbf{s}_T]$, in which every row vector $\mathbf{s}^n = [s^n_1, s^n_2, \cdots, s^n_T]$ is a temporal process on a vertex (cell) $A_n \in \mathcal{A}$, while every column vector $\mathbf{s}_t$ is a graph signal on graph $\mathcal{G}(\mathcal{V}, \mathcal{A})$.

In this context, the prediction problem becomes estimating $\mathbf{s}_t|_{t>T}$, based on observation $\mathbf{S}_{N\times T}$ and average weight matrix $\mathbf{W}$. From exiting literature on GSP [75, 45], we know that, this prediction is possible if process $\{\mathbf{s}_t\}_t$ is JWSS, *i.e.* wide-sense stationary (or weakly stationary) in both vertex and time domains. In the time domain, signal on each vertex can be made stationary by taking the time difference $\Delta\mathbf{s}^K_t = \mathbf{s}_t - \mathbf{s}_{t-K+1}$. So the crux of this prediction problem is the stationarity in the vertex domain, for which we re-examine Eq's. (3.8) to (3.10).

First, we discuss the stationarity of a Case-1 scenario, where only wireless links are utilized to disseminate datum δ. During time step t, if we approximate $\mathbf{p}_{t-1} - \mathbf{s}_{t-1}$ as the summation of a constant vector and a noise vector of Gaussian random variables $\vec{\psi}_t$, the following equation can be obtained for signal $\mathbf{s}_t$:

$$\mathbf{s}_t = [(1-\gamma+\beta)\mathbf{I} + (1-\gamma)\mathbf{W}_t + \beta g(\mathbf{W}_t)]\,\mathbf{s}_{t-1} + \vec{\psi}_t, \tag{3.14}$$

where $g(\mathbf{W}_t)$ is a function of matrix $\mathbf{W}_t$ that maps $\mathbf{W}_t$ into a $0,1$-matrix, and then scaled by some constant c, such that the i,j-th element equals to c if and only if $W^t_{i,j} > 0$.

In a real-world system, the coverage dynamics is observed every K times steps to bridge the frequency gap between data transmission and user mobility (as discussed in Section 3.4.2.1), so we obtain observations $\{\mathbf{s}_{t=jK}\}_{j\in\mathbb{N}}$. Substituting the instantaneous user mobility matrix $\mathbf{W}_t$ with the long-term average weight matrix $\mathbf{W}$, the data-strength vector $\mathbf{s}_{t=jK}$ in Eq. (3.14) can be re-written as $(\mathbf{I} - f(t)\mathbf{W})^j\,\mathbf{s}_0$ plus some random noise, where $f(t)$ is a time-varying

[4] For scenarios with more complicated mobility traces, *e.g.*, a street scenario with both vehicles and pedestrians, or an indoor scenario with sensors and mobile devices, cells of other form may lead to a better result.

coefficient, that can be determined by probability β and γ for every $t = jK$.

Then, process $\{\mathbf{s}_{t=jK}\}_{j\in\mathbb{N}^+}$ is WSS with respect to graph $\mathcal{G}(\mathcal{A}, \mathbf{W})$, because: i) the initial state $\mathbf{s}_0$ and the noise vector are both random, as a result of which both are WSS; ii) any polynomial of $\mathbf{W}$ can be written as a finite product of graph filters [79] defined on $\mathbf{W}$, through which stationarity is maintained; and iii) signal $\mathbf{s}_{t=jK}$ can be seen as a filtered version of $\mathbf{s}_0$. For the ease of notation, we write $\mathbf{s}_t$ for the data-strength observed at time $t \cdot K$ hereafter.

With respect to the more complex Case-2 scenario, where both wireless and wired links are utilized, it is rather difficult to examine the stationarity of $\{\mathbf{s}_{t=jK}\}_{j\in\mathbb{N}^+}$ directly, because of the denominators in the last term of Eq. (3.10), that is, p^n_{t-1} and p^j_{t-1}. However, considering this extra term is in a similar form as the second term in Eq. (3.10), we adopt a 'generalized' weight matrix, *i.e.*,

$$\mathbf{W}^* := (1-\gamma)\mathbf{W}_t + \beta g(\mathbf{W}_t) + \frac{\beta}{\sum_{n=1}^{N} |AP_n|}\mathbf{L}, \tag{3.15}$$

where the number of wired connections from/into each cell is normalized by the total number of APs[5], such that every term in Eq. (3.14) concerning $\mathbf{W}_t$ can be substituted by $\mathbf{W}^*$.

3.5.3 Prediction Framework

Taking advantage of the JWSS property, we present a data coverage prediction framework that consists of a simulator, a pre-processing module, and a predictor, as shown by the block diagram in Figure 3.7. The simulator generates data-strength (signals) $\{\mathbf{s}_t\}_t$ and calculates the average weight matrix $\mathbf{W}$, which are pre-processed into smaller blocks, and then fed into the predictor to estimate the future data coverage.

(1) *Simulator*. The simulator generates signal $\mathbf{S}_{N\times T}$ and the average weight matrix $\mathbf{W}$, based on the `cabspotting` trace [93]. Simulation configurations can be found in Table 3.2.

Table 3.2 Simulation configuration for Section 3.5.

Parameter	Value	Parameter	Value	Parameter	Value
region A	$[0, 6000\ m]^2$	boundary	open	# of experiments	100
# of AP's	231	# of taxis	1072	N	225
area of a cell	400×400	r_c(AP)	200 m	r_c(Taxi)	100 m
prob. β	0.2	prob. γ	0.01	AP conn. prob. (Case-2)	0.5
time interval K	10,50,100,200	time step	1 s	time-span T	12000

(2) *Pre-processing* This component partitions graph $\mathcal{G}$ into smaller subgraphs, as shown in the example in Figure 3.8a, to avoid heavy computation induced by frequent eigen-decomposition

[5] In this case, we still have $\mathbf{s}_t = [\mathbf{I} + f(t)W^*]\mathbf{s}_{t-1} + \vec{\psi}$, but with a deviation caused by the normalization over the total number of APs, instead of the time-varying p^n_{t-1} in the denominators. To compensate this effect, we adjust the graph partition (discussed in the following Section 3.5.3) with historic signal observations, when high discrepancies between the observation and prediction occur.

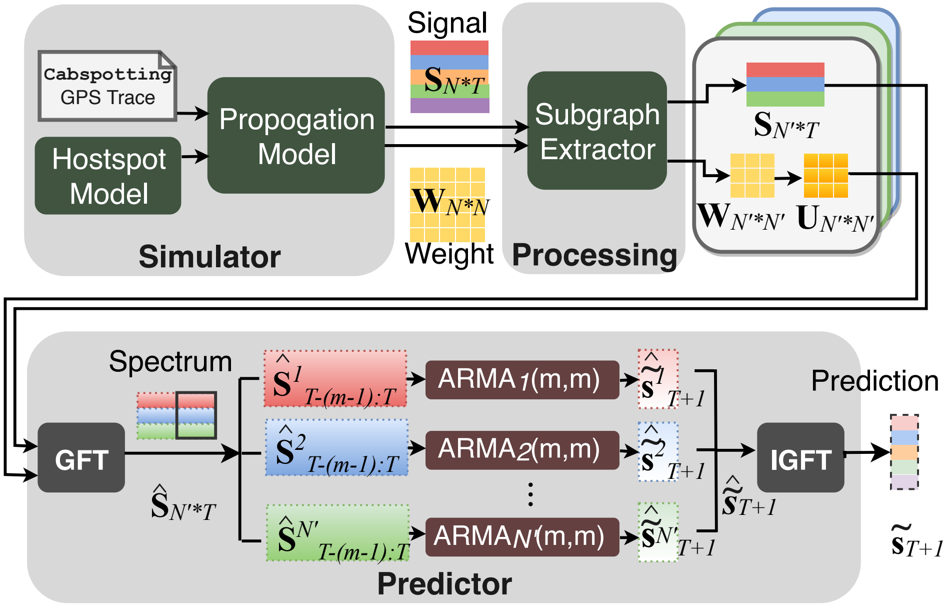
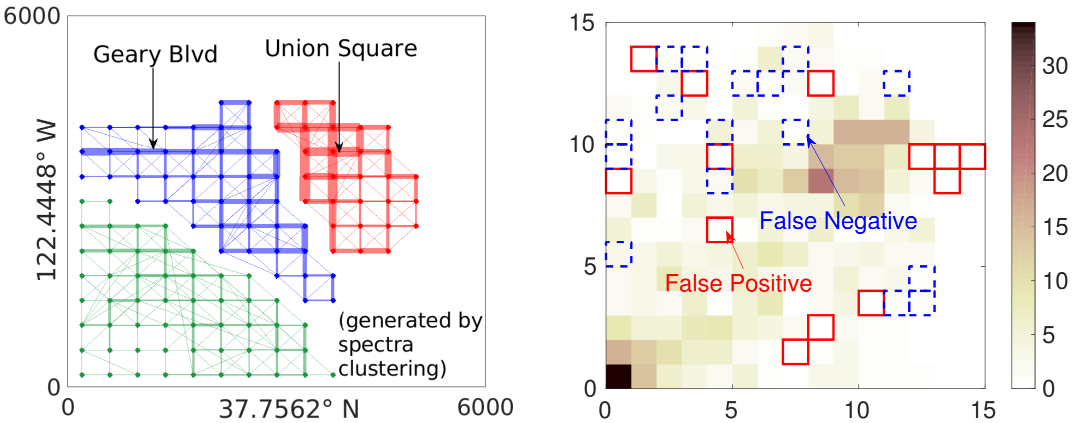

Cabspotting GPS Trace
Hostspot Model
Propogation Model
Simulator
Signal
$\mathbf{S}_{N*T}$
$\mathbf{W}_{N*N}$
Weight
Subgraph Extractor
Processing
$\mathbf{S}_{N'*T}$
$\mathbf{W}_{N'*N'}$
$\mathbf{U}_{N'*N'}$
Spectrum
GFT
$\hat{\mathbf{S}}_{N'*T}$
$\hat{\mathbf{S}}^{1}_{T-(m-1):T}$
$\hat{\mathbf{S}}^{2}_{T-(m-1):T}$
$\hat{\mathbf{S}}^{N'}_{T-(m-1):T}$
ARMA1(m,m)
ARMA2(m,m)
ARMAN'(m,m)
$\hat{\tilde{\mathbf{s}}}^{1}_{T+1}$
$\hat{\tilde{\mathbf{s}}}^{2}_{T+1}$
$\hat{\tilde{\mathbf{s}}}^{N'}_{T+1}$
$\hat{\tilde{\mathbf{s}}}_{T+1}$
IGFT
Prediction
$\tilde{\mathbf{s}}_{T+1}$
Predictor

Figure 3.7 Block diagram of the prediction framework, in which the prediction process is illustrated.

of large matrices, as well as the case of disconnected $\mathcal{G}$, due to absence of movement across cells. Note the JWSS property is sustained on subgraphs, especially when K is small, because only a few neighboring cells are relevant in the short-term evolution of the dynamics. We consider two graph partition measures: a *fixed* partition of $\mathcal{G}$ into 10 subgraphs based on spectra-clustering, and a *dynamic* partition of $\mathcal{G}$ that takes a short ($< 10\%$) history of signal, in this case the first $0.1\frac{T}{K}$ observations, into consideration. Note that the dynamic partitioning scheme is not adaptive, which means the graph $\mathcal{G}$ is partitioned only once, to avoid heavy computation.

(3) *Predictor.* The predictor works on the sub-graph (of size N') level, with (sub-)weight matrix $\mathbf{W}_{N'\times N'}$. It first computes the GFT of signal portion $\mathbf{S}_{N'\times T}$, then feeds each line of the spectrum $\hat{\mathbf{S}}^{i}_{T-(m-1):T}$ to an auto regressive moving average (ARMA)(m,m) model trained with history observations, to generate the forecast of GFT spectrum $\hat{\tilde{\mathbf{s}}}_{T+1}$. Finally IGFT is applied to obtain prediction $\tilde{\mathbf{s}}_{T+1}$, which is the estimated $T+1$-th data-strength, for the observation at time $t = (T+1)K$. We choose $m = 2$, because we found through experiments that, despite heavier computation, larger m does not increase accuracy. Data coverage for every observation is then determined by comparing $\tilde{\mathbf{s}}_{T+1}$ with the threshold 0.5.

The rationale behind the prediction is that, when a time-vertex process $\{\mathbf{s}_t\}_t$ is JWSS, the (future) GFT spectrum of the signal along every frequency (eigenvalues of the graph Laplacian $\mathcal{L}_{\mathcal{G}}$) can be *independently* predicted [75]. In this sense, each row of the spectrum is a time series, which can be predicted by statistical methods like ARMA, or other machine learning methods, such as recurrent neural network (RNN), and long short-term memory (LSTM) model. ARMA is adopted in this work, because in such single-variate short-term prediction problem with relatively few training data, statistical methods generally out-performs the latter in accuracy, complexity, and training time [77].



(a) Examples of subgraphs from partitioning. **(b)** An example of prediction result.

Figure 3.8 Examples of pre-processing and prediction result (color indicates real signal value).

3.5.4 Results and Discussion

In the validation, we consider time intervals of different lengths K (10s, 50s, 100s and 200s) to validate the proposed model and prediction framework for different applications. For instance, prediction with finer time granularity is more beneficial to ITS systems, in which traffic information has more stringent delay requirements, while prediction at a coarser time grain, but less computationally expensive, should satisfy the need of mobile advertisement applications.

3.5.4.1 Summary of Prediction Accuracy

Performances of the prediction framework are summarized in Table. 3.3, in which the error ratio (percentage of time instances with erroneous predictions) is calculated for each cell, and averaged over 100 experiments. Erroneous prediction includes false positive when a ‘covered’ prediction is made for a not covered cell, and false negative when the opposite happens, as shown in Figure 3.8b. We list four observations from the result summary:

(1) False positives are more prevalent in fixed partition due to ineffective division of the graph, while false negatives are prevalent in dynamic partition, because the impact of data forwarding actions (and that $\beta >> \gamma$) is not fully regarded by the average weight matrix $\mathbf{W}$, compared to mobility. This is especially true when the observation/prediction intervals is long, *e.g.*, $K = 100$ and $K = 200$.

(2) There is a clear decrease in prediction error when finer time granularity (smaller K) is employed, because: i) the ARMA model is better trained with more observations; ii) Stationary of the difference signal $\Delta \mathbf{s}_t^K$ is better preserved, when extended from one-step evolution in Eq. (3.14). This effect in in conjunction with the false negative errors listed in (1), because mobility is fully captured by the model for longer intervals, while the cumulative error induced by not fully incorporating data transmission is severer.

(3) Extending the weight matrix $\mathbf{W}$, which only considers user mobility, to the generalized

Table 3.3 Summary of prediction error for different settings.

Time interval K (s)	**False Positive** (%)		**False Negative** (%)	
	fixed(f)	dynamic(d)	fixed(f)	dynamic(d)
	Case-1: Wireless Connection Scenario			
200	13.77	8.39	10.80	10.93
100	13.40	8.04	10.36	10.43
50	12.51	6.74	10.08	9.72
10	10.05	3.98	7.28	5.87
	Case-2: Heterogeneous Connection Scenario			
200	12.18	7.73	11.13	10.79
100	12.18	7.54	10.62	10.40
50	11.18	6.54	10.20	9.80
10	8.58	4.61	7.45	6.54

form $\mathbf{W}^*$, which jointly considers mobility, wireless data transmission across cell borders, as well as wired connection between APs, does not affect the prediction accuracy much for the Case-2 scenario, as can be observed from the comparison between the two cases in Table 3.3.

(4) Dynamic graph partition can effectively reduce false positives, compared to the fixed scheme, but such effect is not as obvious for false negatives. To understand this, we further examine the geographical distribution pattern of prediction errors.

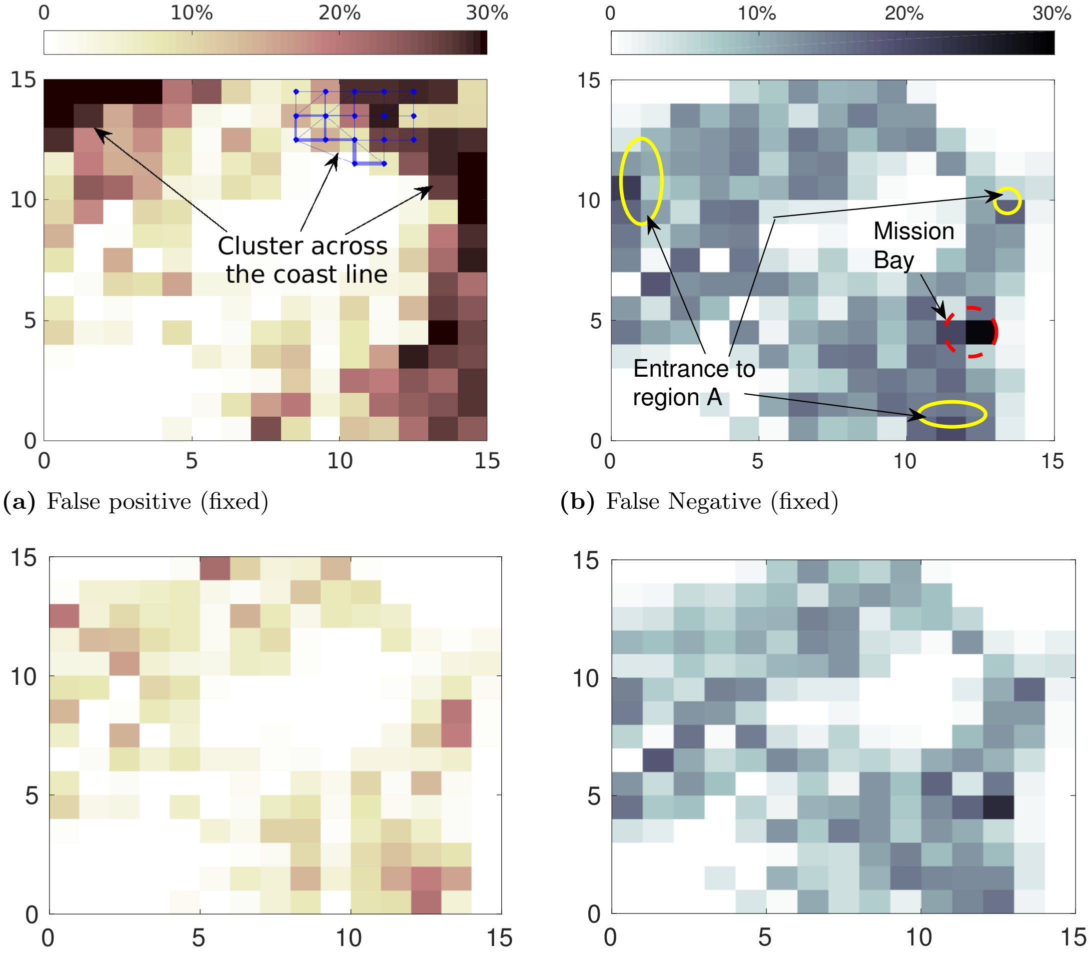
3.5.4.2 Accuracy with Respect to Location

Figure 3.9 and Figure 3.10 illustrate the average prediction error of each cell, for Case-1 and Case-2 scenarios, respectively. The darker the color of a cell, the higher the ratio of prediction error. As can been seen from both figures, there is a clear pattern with respect to location.

(1) *False positives* are most severe at the top-left and top-right corners in Figure 3.9a, which are either across the coastline (top-right), or close to it (top-left). In the fixed graph partition, some subgraphs expand over the coast line, as indicated by the blue cluster at the top-right corner in Figure 3.9a. This can be easily identified and corrected by the dynamic partition scheme (as in Figure 3.9c), which breaks these subgraphs based on historical observations, as shown in Figure 3.9c, for the Case-1 scenario, and Figure 3.10a, for the Case-2 scenario.

However, as we emphasis on the impact of wired connections in Case-2, which successfully improves the accuracy for cells with APs (bottom left and middle part of the region), the top and right 'boarder' cells are left with more false positives, as indicated by the darker cells in Figure 3.10a. This increase in false positives is due to the over-prediction introduced by MFA, which is especially prevalent when a cell has very few APs, or is rarely visited by taxis (*e.g.*, dark cells at the right fringe, in Figure 3.10a), such that the 'well-mixing' assumption of MFA does not hold any more.

(2) *False negatives* outline the terrestrial boundaries of the region, among which the most severe errors occur at two sets of locations, highway entrance to region A (marked by the solid



(a) False positive (fixed)

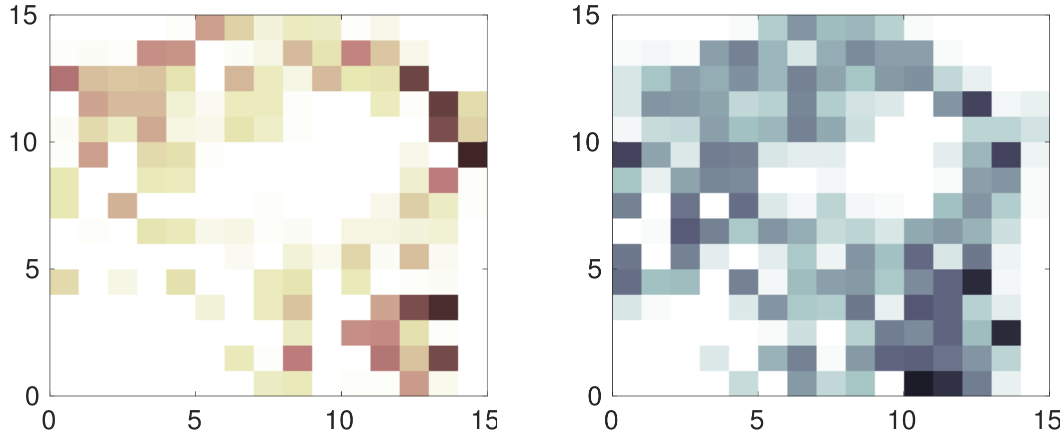
(b) False Negative (fixed)

(c) False Positive (dynamic)

(d) False Negative (dynamic)

Figure 3.9 Prediction error of each cell in a Case-1 (wireless links only) scenario.

yellow circle in Figure 3.9b), and the Mission Bay area (identified by the dashed red circle in Figure 3.9b). High error ratio at the highway entrances is caused by data-carrying entities entering the region through these cells, because we assume the boundary of region A is open, as a result of which data dissemination can actually happen outside region A, as discussed in Section 3.5.1.2. For the latter case with respect to the Mission Bay area, a narrow water hinders taxis from moving across cells (low weight in matrix $\mathbf{W}$), but can not prevent data forwarding over the air, which creates a discrepancy between weight $\mathbf{W}$ and the intensity of cross-cell data forwarding actions. This can not be effectively improved by dynamic partitioning, as shown in Figure 3.9d. Similar effect is also seen in Figure 3.10b, where the inaccurate predictions mainly reside at the entrance/exit of region A. To further eliminate the false negative errors in these sub-regions, we expect an adaptive dynamic graph partition can properly incorporate the impact of cross-cell data forwarding, but is much more computationally expensive, as it requires graph partition (and eigen-decomposition) at every observation/prediction.



(a) False positive (dynamic)

(b) False negative (dynamic)

Figure 3.10 Prediction error of each cell in a Case-2 (wireless and wired links) scenario.

3.6 Summary

In this chapter, we raise the question of *where data are*, in heterogeneous edge networks, which is becoming the main scenario for data dissemination services. To address the *whereabouts* of mobile data, we formally define the scope, entity model, and data coverage for a data dissemination process in such networks, propose a data-strength metric such that data coverage can be quantified, and formulate the data coverage dynamics as a time-vertex process of numeric graph signals. This formulation enables us to observe, analyze, and predict the dynamic evolution of coverage dynamics, en route of data movements. Specifically, from snapshots of the coverage dynamics, we observe the impact of user mobility is high, especially when entities are less active forwarding or the maximum moving speed is low; as the maximum speed increases, such impact decreases but is still lower bounded. From this observation, we propose to predict data coverage based on system-level user mobility traces, which can be easily obtained without raising privacy concerns. Simulation with real-world mobility traces shows that our framework is a practical solution for predicting data coverage in heterogeneous wireless networks.

Chapter 4

Governing Rules: Modeling and Analysis of Task Offloading Processes in the Fog

In the previous chapters, we discussed the dissemination process of *one* data block, particularly its lifetime and whereabouts, in wireless networks. As data services quickly migrate to the network edge, which is composed of numerous resource-constrained wireless devices and access network elements, the dissemination processes of *multiple* data blocks may interfere with each other, due to their competition for resources. A typical example is the emerging *fog* paradigm, in which a *task*, that is, a computation-intensive (and delay-sensitive) data block [139], can be offloaded to resourceful fog nodes nearby, in order to reduce latency, or save energy. The offloading process is affected by the amount of accessible resource in the network, and its execution in turn affects how other tasks will be offloaded, making it difficult to evaluate the performance of such large-scale fog systems. To address this open problem, we study the task offloading process in terms of *how multiple data blocks move* in this chapter. We propose a gravity-based offloading model, to describe how the target fog node is selected under different offloading criteria. This model allows us to evaluate the performance of a large-scale fog system, particularly how much (local and shared) resource such offloading processes consume, which are quantified by the *device effort* and *network effort* metrics. With respect to the scalability of fog, which is a major design concern of the paradigm, we find that, under the gravity-based offloading rule, the lifetime of an individual task does not deteriorate as the network size increases, while the total resource needed for the system scales linearly. As our model can describe various offloading criteria, these results are generally applicable in understanding how data move in a fog network, and the amount of resource needed for such systems.

4.1 Introduction

With the proliferation of smart devices [120, 119], including smart phones, wearables, and sensors, Internet-of-Things (IoT) is driving a digital transformation in all aspects of modern life [4]. As the vision quickly turns into reality [90], IoT brings about new challenges to its provision network, including stringent latency requirement, resource-limited devices, and the prohibitive scale. These requirements are difficult to satisfy with the existing cloud computing paradigm, due to it centralized structure, where cloud servers are usually deployed in privately-owned, remote data centers. Consequently, *fog computing* is introduced [119] and envisioned as a promising paradigm in IoT provisioning [4].

4.1.1 Fog Emerges on the Edge: Remedy or Resource Drain?

The principle feature of the fog paradigm is bringing data services to the *edge* of the network, such that the processing and dissemination of data are available, right where the generation and consumption of data take place. Instead of drawing the surging mobile data traffic to cloud servers through the core network, data are processed by (less powerful) fog servers deployed in edge devices, *e.g.*, commercial edge routers, Cloudlet and IOx [136], and even mobile devices [49], which are referred to as *fog nodes*. In this sense, fog computing is a broad concept that includes mobile edge computing (MEC), mobile cloud computing (MCC), cloudlet, and mobile ad hoc cloud computing (MACC) [139]. From the perspective of network architecture, some researchers view fog nodes as an intermediate layer between "things" in the IoT and remote servers in the cloud [2, 49], while others view it as a fully distributed substitute of the cloud architecture [28, 118]. Despite the different opinions in the fog architecture, a consensus for the fog continuum [119] is that, its monumental purpose is to provide data offloading options, with two design objectives [2] tailored for IoT applications:

(1) *Short Delay.* Per request of real-time applications in IoT, such as smart infrastructure, autonomous driving, and virtual assistance [90], a fog system is expected to reduce the service latency to the ms level (< 10 ms), comparing to the 30$\sim$100 ms in the current cloud computing paradigm. This demand requires the fog paradigm to maintain a sufficient number of fog servers in the vicinity of any *task node*, which can be any wireless device seeking to offload data.

(2) *Scalability.* Due to the massive scale of the geo-dispersed IoT systems [4], a fog system is expected to process tasks offloaded from a large number of task nodes, *e.g.*, the vast amount of sensors, cameras, and smart vehicles in an intelligent transportation systems (ITS). Considering the frequent changes of locations, connectivity and offloading criteria, this objective requires the fog paradigm to scale easily, *i.e.*, operating with simple organization and minimum overhead. In fact, openness and scalability have been recognized as the major design concerns for the fog paradigm, by the OpenFog consortium in the IEEE OpenFog reference architecture [2].

With these objectives fulfilled, the fog paradigm seems a good remedy for the large-scale delay-sensitive IoT applications, as it rallies up resource at the network edge, and takes advantage of the huge numbers of candidate fog nodes (routers, AP, and mobile devices) in a distributed manner. However, as mobile traffic continues to grow [120], will the resource-limited network edge be able to keep such a fog eco-system afloat? In other words, *how to evaluate the performance of the fog paradigm* with respect to the massive task offloading processes?

Answer to this question helps us understand the impact of data services, which is non-trivial for the edge network, a heterogeneous wireless system. First, from the perspective of service provisioning, resource constraint has been a major issue at the network edge [78], especially for communication resources. While processing and storage can be boosted by simply deploying more fog nodes, communication resource, including spectrum, time, code, and space, is *shared* by the system, and hence subject to further degeneration as the system scales. Nonetheless, fog computing allows power-constrained end devices to save energy by avoiding local processing, at the cost of frequent short-range communications with nearby fog nodes, which further aggravates the spectrum scarcity concern [13]. Second, from the perspective of traffic pattern, IoT applications are so broad and complex, that data are generated in various volume, form, and frequency, *e.g.*, there are periodic, short sensory data from smart meters, as well as spontaneous, long video clips from ground traffic monitoring. As a result, the fog paradigm, which provides infrastructure-as-a-service (IaaS) and platform-as-a-service (PaaS) to IoT, impose unpredictable resource demand on the underlying edge network. For such a large-scale system that is still in the design stage, it is essential to understand its impact on existing networks, and its performance as the system scales, before it can be deployed.

4.1.2 Related Work

As am exemplified application in the fog, data offloading has been extensively studied in different contexts, including MCC, MEC, and cloudlets, from the aspects of synchronized data storage [123], bench-marking [49], deployment [101], and so on. Considering that fog is specifically designed for IoT applications, in which data blocks are generated by resource-limited nodes, and need to be processed within a time bound, we focus on the offloading process of computation-intensive data blocks [139], *i.e.*, *tasks*.

The central question in task offloading is, for each individual task, whether a task should be offloaded, and if so, to which fog node, both of which are determined by the offloading mechanism. Existing literature on this topic can be categorized into centralized and distributed schemes, typical examples of which are shown in Table 4.1. The former family (the majority) relies on a single centralized entity to determine where each task should be sent, *e.g.*, [24], which is against the open and distributed nature of fog computing, so we focus on the distributed schemes, which further branches into cooperative and non-cooperative types.

In cooperative offloading schemes, *e.g.*, [140, 134, 135, 23, 97], fog nodes exchange information to optimize resource allocation, or collectively schedule the offloaded tasks, to minimize the offloading latency [133] and/or energy consumption [129] for a single task. In other words, fog nodes cooperate to determine the offloading target. Though cooperation among fog nodes are highly beneficial to Quality-of-Service (QoS) provisioning and load-balancing, it is less likely in a large-scale fog continuum, due to the prohibitive communication overhead, lack of interoperability among different providers, and security concerns of end users. In this regard, Tang and He recognizance the competition among fog nodes, modeled the offloading process as a non-cooperative game [113] to study the impact of users' behavioral biases, and evaluated their results with small-scale experiments. In addition, considering the wireless link may not be stable during the task offloading, Wu considered fault recovery in the offloading process, and aimed to maximize revenue of the network [128].

Table 4.1 Fog node selection schemes for task offloading.

Example scheme	Selected by	Objective	Cooperative	Approach
[24]	central control	delay	yes	optimization
[133]	fog node	delay	yes	optimization
[129]	fixed	QoE and energy	yes	optimization
[128]	task node	fault recovery	no	queuing theory
[55]	task node	energy	no	optimization
[138]	task node	delay	no	heuristic scheme
[113]	task node	subjective choice	no	game theory

An open, yet challenging question is, *how to evaluate the performance of such fog systems*? To be more specific, what is the expected processing time for a task, and how much resource will be consumed by the fog-based IoT system? Considering that different offloading schemes can be adopted, and they differ in many aspects, as shown in Table 4.1, it is difficult, if not impossible, to measure the system performance by experiments. This difficulty can be further exacerbated by the large, and growing size of the fog-based IoT system, which is expected to cater for millions of IoT devices [4]. Consequently, the *scalability* of fog computing, as identified in the survey [139] on fog computing, is the crux of the performance evaluation problem. On one hand, the increase in network size permits more offloading options, which increases the offloading probability, and hence the QoS of end applications. On the other hand, frequent data transmission as a result of higher offloading probabilities raise concerns in resource consumption. In addition, some participants in the fog, *e.g.*, smart phones, may act as both task node and fog node, such that the total amount of data (to offload) also increases as the system becomes larger. Meanwhile, task nodes can have their own offloading criterion/algorithms, as a result of which the impact of network size is unclear.

4.1.3 Our Approach and Contributions

Seemingly simple, performance evaluation of task offloading is very challenging in the fog computing context: First and foremost, the distributed and open nature of fog introduces heterogeneity in offloading criteria. Consequently, it is difficult to describe actions of individual nodes, and hence more difficult to capture behavior of the system as a whole. Second, with respect to resource need, particularly scalability, experiment is not an easy option, because it is impossible to implement and measure all combinations of offloading+communication schemes in the envisioned fog system, which may not even be readily available. Therefore, we need a generic model and a set of universal metrics that can accurately depict data offloading processes, with flexible offloading criteria, yet manageable to derive scalability properties of the fog paradigm.

Instead of focusing on inter-node coordination and trade-offs, we address the performance evaluation problem from the perspective of data, and take a completely different approach, that is, finding out *how data blocks move* in the fog during the offloading process. The rationale behind this approach is: resource consumption, in forms of spectrum, time, and storage space, are all driven by movements of data blocks, and can be examined through the traversing paths of tasks as data blocks. In regard to task offloading process in the fog, which is a typical example of multiple data blocks move under the influence of each other, our contributions can be summarized as follows.

We propose a gravity-based offloading model that views tasks as particles, whose movements are driven by the *gravity force* between the particle and different fog nodes. Moreover, a data block moves toward a fog node with a probability proportionate to the gravity force, allowing our model to capture the adjustments of offloading target, either due to load-balancing among fog nodes, or subjective preferences of the task node. Then the impact of data offloading processes can be viewed as the effort of the gravity force, in the form of data movements.

We define evaluation metrics, namely, *task lifetime*, *device effort* and *network effort*, which quantify the end-to-end delay of a task, the consumption of storage/processing resource at the device level, and that of communication/coordination resource at the network level, respectively.

We find that, under a generic gravity rule, which applies for distance, energy, and delay-based offloading criteria, the lifetime of a task remains constant (when the system is lightly loaded), or decreases (when the processing load is heavier) with respect to the network size, and both the device and network efforts of the system scale linearly. This result shows though the QoS of individual tasks can be guaranteed as the network scales, the total network resource consumption may exceeds what the system can provide.

4.2 System Model and Problem Formulation

In this section, we formally introduce the terminology and settings on nodes, tasks, and the offloading process, define the performance metrics from the perspective of data movements, and formulate the offloading problem.

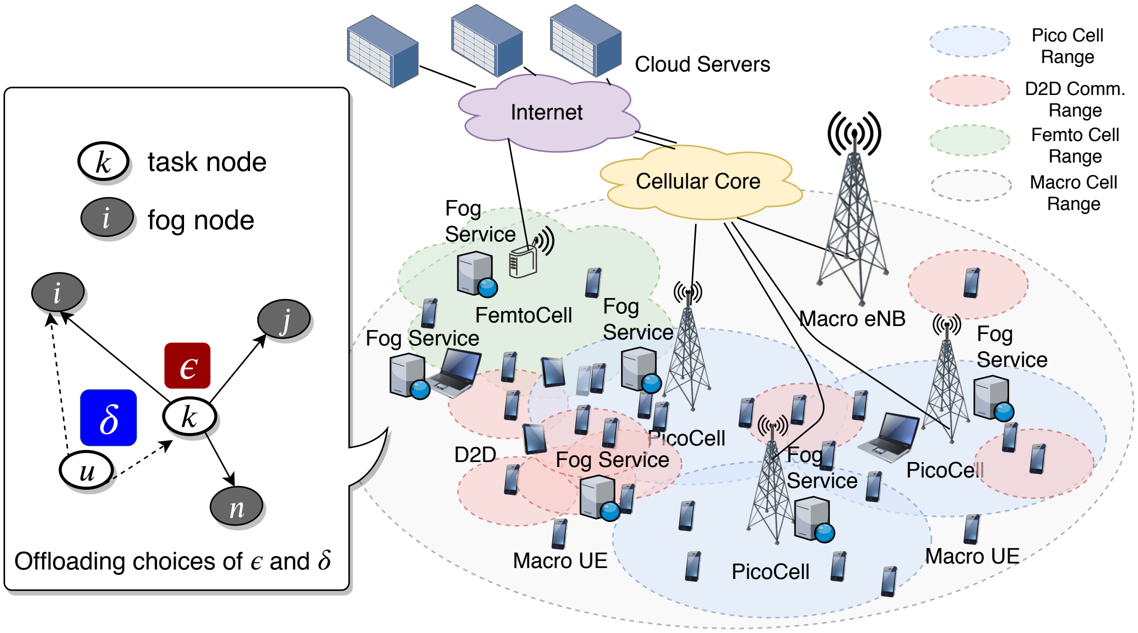


Figure 4.1 An example of a fog system hosted by a heterogeneous wireless network, and a data offloading scenario (inner box on the left) in the fog.

4.2.1 Network Model

Consider a fog system hosted by the 5G cellular network and wireless local area network (LAN), a miniature example of which is illustrated in the right of Figure 4.1. In this network[1], the fog service is provided by BS's (of picocells and femtocells), AP's, and mobile devices (including smart phones, tablets, laptops, and so on), which we refer to as *fog nodes*. Any device that is authorized to offload tasks (to fog nodes) is referred to as a *task node*, so it is possible that a node (device) can act as a fog node and a task node at the same time. For instance, an end device k can choose to offload a heavy video trans-coding task ϵ to nearby fog servers deployed at the AP j, while accepting a file offloading (further to the cloud) task δ from a nearby IoT sensor u, as shown in the zoomed box on the left of Figure 4.1.

Therefore, we consider a node $n \in E$ to be composed of two components: a fog node, which is associated with a CPU with processing capacity c_n, and a first-in-first-out (FIFO) queue; and a task node, which generates tasks at the rate of β. As a fog node that can process tasks, n announces its capacity c_n within its transmission range r_n, while as a task node, n decides whether, and to whom, every of its generated tasks will be offloaded.

Considering any node may move in time span $T = [0, \infty)$, for a node e, let $\{X_e(t)\}_{t \in T}$ denote its trajectory over time, where each $X_e(t) \in \mathbb{R}^2$ is a vector on a 2-D plane representing

[1] Considering resource bottlenecks mainly exist in wireless networks, in this chapter, we restrict the scope of the fog paradigm to the network edge, despite the fact that wired connections between fog nodes and cloud servers are also viewed as part of the fog system in some research, *e.g.*, [133, 2].

the location of node e at time t. For a node $e \in E$, located at $X_e(t)$ at time t, denote its *neighbors* as

$$N_e(t) = \{i \in E \mid d(X_i(t), X_e(t)) \leq \min\{r_e, r_i\}\}, \tag{4.1}$$

where r_e is the transmission range of node e, beyond which data transmission is not possible.

Suppose node e utilizes a wireless channel of bandwidth BW (Hz) to communicate with node $n \in N_e(t)$, located at $X_n(t)$, where the noise power measures at P_0 (dBm) inside this channel bandwidth. Then if node e with transmission power P_e (dBm), which remains the same throughout the data dissemination process, the data transmission rate from node e to node n can be obtained as

$$\begin{aligned} R_{e\to n}(t) &\leq BW \log_2(1 + \mathrm{SNR}) \\ &\leq BW \log_2(1 + 10^{\frac{1}{10}(\mathrm{RSSI}_n(\mathrm{dBm}) - \mathrm{Noise}(\mathrm{dBm}))}) \\ &\simeq \frac{BW(\log_2 10)}{10}\left[P_e - P_0 - \kappa d(X_n(t), X_e(t))\right] \end{aligned} \tag{4.2}$$

where κ is the path-loss exponent in the current system, and $d(X_n(t), X_e(t))$ measures the distance between node e and node n at time t. The transmission range r_e can then be defined as $r_e := \sup\{d > 0 \mid \frac{BW(\log_2 10)}{10}(P_e - P_0 - \kappa d) \geq R_{min}\}$, where R_{min} is the minimum possible transmission rate that is allowed in a RAT.

Remark 4.1. *For fair comparison and simplicity reasons, we assume that the communication between any pair of nodes follows the same set of parameters, including transmission power P_e, noise level P_0, and channel bandwidth BW, such that the transmission rate from node e to node n can be simplified into the following form,*

$$R_{e\to n}(t) \simeq R_{max} - 0.32BW\kappa d(X_n(t), X_e(t)) \overset{\triangle}{=} R_{max} - \psi d(X_n(t), X_e(t)), \tag{4.3}$$

where $R_{max} \simeq 0.32BW(P_e - P_0)$, and constant coefficient $\psi = 0.32BW\kappa$.

Writing data rate $R_{e\to n}$ as a linear function of distance $d(X_n(t), X_e(t))$ is reasonable for several different communication schemes. For instance, this linear relationship is observed in measurements of WiFi signals, [39]. As for the parameter R_{max}, *i.e.*, the peak data rate, we survey measurements in real-world networks for comprehension in a numeric sense: The peak data rate R_{max} in WiFi systems measures at 8.3 Mbps unlink and 27.33 Mbps downlink for mobile devices, and 32.88 Mbps unlink and 96.25 Mbps downlink for fixed devices [109]. The peak data rate in LTE measures at 17.2 Mbps unlink and 42.4 Mbps downlink [41]. For the future 5G communication, or WiFi with mmWave, the peak data rate can reach 5 to 40 Gbps [52] in lab environments.

4.2.2 Task Model

Consider task ϵ, which is captured as a seven-element tuple $(t_\epsilon, L_\epsilon, \tau_\epsilon, S_\epsilon, D_\epsilon, \phi_\epsilon, \alpha_\epsilon)$, whose ranges and physical meaning can be found in Table 4.2.

Table 4.2 Attributes of a task ϵ (determined at t_ϵ).

Attribute	Range	Physical meaning
t_ϵ	$\mathbb{R}^+$	the time instant ϵ is generated
L_ϵ	$\mathbb{N}^+$	volume of task ϵ (bits)
τ_ϵ	$\mathbb{R}^+$	timeout/delay bound
S_ϵ	E	task node who generated ϵ
D_ϵ	E	target fog node to offload
ϕ_ϵ	$[100, 400]$	processing intensity (CPU cyles/bit)
α_ϵ	$(0, 1]$	coefficient of volume change after processing

Among these attributes in Table 4.2, the first five are common attributes of data blocks in any dissemination process, while the last two capture the characteristics of data blocks as tasks to be offloaded. For example, a file ϵ of size L_ϵ, to be transferred from the source node $S_\epsilon = k$ to the destination node $D_\epsilon = n$, needs to be completed by time instant $t_\epsilon + \tau_\epsilon$. As a task to be offloaded to node D_ϵ, data block ϵ has two more attributes in addition to the basic model, *i.e.*, the processing intensity ϕ_ϵ and volume change coefficient $\alpha_\epsilon \in (0, 1]$, because a task ϵ needs to *processed* for an expected result $\alpha(\epsilon)$, where function $\alpha : \{0, 1\}^{L_\epsilon} \to \{0, 1\}^{\alpha_\epsilon L_\epsilon}$ describes the processing procedure that maps task δ to the result $\alpha(\epsilon)$. For example, a video trans-coding task ϵ to compress an 1 minute 1080p video clip to a low-resolution 360p version has data volume $L_\epsilon = 165$ Mb, processing intensity $\phi_\epsilon \simeq 290$ cycles/bit (calculated from data in [9, 73]), and volume change coefficient $\alpha_\epsilon = 0.33$, as the processing result $\alpha(\epsilon)$, *i.e.*, the 1 minute 360p video clip, is only 55 Mb in volume.

Three attributes of the task tuple, *i.e.*, volume L_ϵ, processing intensity ϕ_ϵ, and volume change coefficient α_ϵ, may differ considerably among different types of tasks, *e.g.*, a text-translation task have a small L_ϵ on the level of Kb, and coefficient $\alpha_\epsilon \simeq 1$, because the size of the translated file will be comparable with that of the original text file, both of which are small; virtual reality (VR)/augmented reality (AR) tasks have large L_ϵ, and coefficient $\alpha_\epsilon \simeq 1$, because the major processing object are images, which are relatively large in volume; while object recognition and cloud offloading tasks have medium L_ϵ, and small coefficient $\alpha_\epsilon \simeq 0$, because the expected result $\alpha(\epsilon)$ will be much shorter compared to the original data block ϵ.

All the attributes of task ϵ are determined at its generation (*i.e.*, at time t_ϵ). Among these, the offloading target D_ϵ decides the direction of ϵ's movements during the offloading, which is determined by the data owner/generator with an *offloading criterion*, taking into consideration the current status of the network, *e.g.*, reachability, processing capacity, and cost of the fog

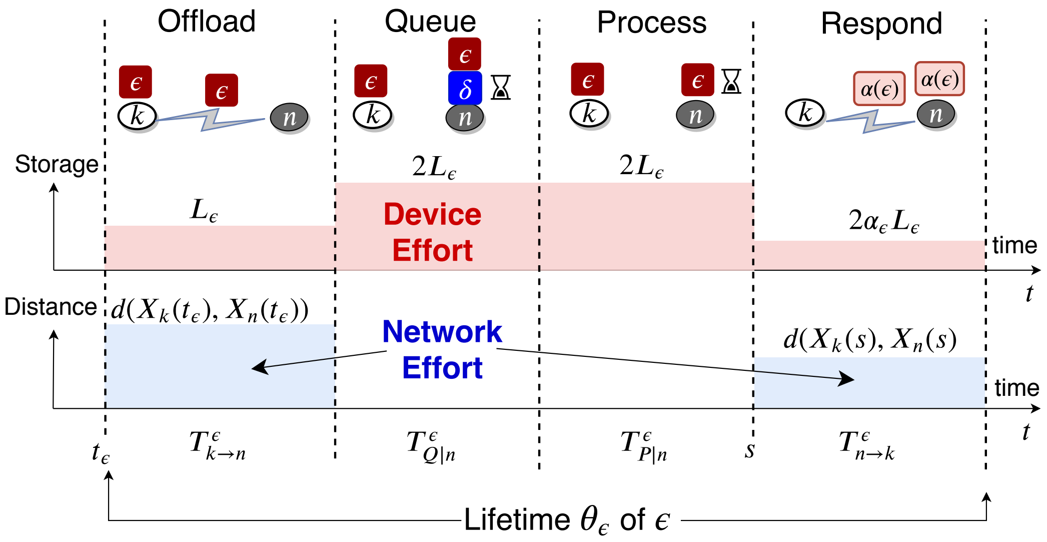


Figure 4.2 Efforts spent to offload task ϵ from task node k to fog node n.

service. For instance, in the offloading scenario shown in Figure 4.1, task node k can choose fog node n as the target fog node, where ϵ will be offloaded to, by a distance-based offloading criterion to save energy; and task node u may choose fog node i, according to a delay-based offloading criterion to reduce service latency. To address how the target is selected, we propose a *gravity* model in the next Section 4.3, which can describe a family of task offloading criteria.

4.2.3 Performance Metrics through Data Movements

The offloading process of task ϵ, is a course of transporting data block ϵ from its generator, task node k, to a fog node n, getting processed at n, and then returning the result $\alpha(\epsilon)$ back to its generator k. This process can be described by four stages: offloading (from k to n), queuing (at n), processing (at n), and responding (from n back to k), as shown at the top of Figure 4.2. Note that the offloading target, *i.e.*, fog node D_ϵ, is determined by the task node $S_\epsilon = k$ at t_ϵ, and can take value $D_\epsilon = k$, which means k decides to process ϵ locally. In this case (local processing), the offloading process only contains the queuing and processing stage.

We consider performance[2] of task offloading processes from two aspects: the completion time of individual tasks, and the resource consumption during the offloading process. From the perspective of individual users, the generator S_ϵ of a task ϵ needs to know the time that task ϵ is completed, which is defined as the *lifetime* of task ϵ in the fog. From the perspective of the fog system, we define *device effort* and *network effort* to quantify the storage, and communication resource consumption, respective.

Let $\mathcal{H}_\epsilon(t) \subset E$ denote the set of nodes that has data block ϵ locally stored at time t. Then

[2] Though task offloading in the fog paradigm is usually one-hop, *i.e.*, direct transmission between the task node S_ϵ and a fog node D_ϵ, we define the performance metrics in a more general sense, such that they also apply to multi-hop scenarios.

with respect to ϵ, the state of each node $e \in E$ is a random variable $\epsilon_e(t) = \mathbb{1}_{\mathcal{H}_\epsilon(t)}(e) : \Omega \to \{0,1\}$ in probability space $(\Omega, \mathcal{F}, \mathbb{P})$, and changes in $\mathcal{H}_\epsilon(t)$ can be represented by the random process $\{\vec{\epsilon}(t)\}_{t\in T}$ of random vectors $\vec{\epsilon}(t) = \{\epsilon_e(t)\}_{e\in E}$. For a generic data dissemination process in wireless networks, the movement trace of data block ϵ can be described by a process of random elements, $\{\mathcal{X}(t)\}_{t\in T}$, where $\mathcal{X}(t) = \bigcup_{e\in\mathcal{H}_\epsilon(t)}\{X_e(t)\}$, as a result of actions taken by the carrying nodes $\mathcal{H}_\epsilon(t)$. The movements of data block ϵ will stop when its *lifetime* θ_ϵ is reached.

Definition 4.1. *Consider task ϵ, which is generated by task node $S_\epsilon \in E$ at time t_ϵ. Let $\alpha(\epsilon)$ denote the expected processing result of task ϵ. The lifetime of ϵ is define as the time between its generation, and the time that the last bit of $\alpha(\epsilon)$ reaches the task node S_ϵ, that is,*

$$\theta_\epsilon := \min\big\{\tau_\epsilon, \inf\{t \geq 0 \mid S_\epsilon \in \mathcal{H}_{\alpha(\epsilon)}(t)\} - t_\epsilon\big\}, \tag{4.4}$$

where τ_ϵ is the timeout bound of task ϵ.

Lifetime measures the minimum period of time that task ϵ (and its copies) is physically present in the fog system. It is upper bounded by the timeout τ_ϵ (an attribute of the task determined at generation) of that data block, after which time the existence of ϵ is no longer necessary, though some carrying nodes of this data block can still hold to it longer, at their own will. In the offloading process (before the end of its lifetime), movements of the task as a data block, are sustained at the cost of *efforts* by the fog system. Three types of resource are consumed to complete a task: communication, storage, and processing. Considering that the amount of processing resource needed is fixed, and can be determined by the product of volume L_ϵ and processing intensity ϕ_ϵ, we focus on the first two that are changing over time.

Remark 4.2. *The concept of* ***effort*** *can be understood with an analogy in physics. If we view a data block ϵ as a solid object with mass $\phi_\epsilon L_\epsilon$, which is initially static at its generator S_ϵ. In this case, external forces have to be applied onto the data block to change its motion (from static to moving), which is accompanied with work by the force, spent in the form of energy.*

During the lifetime of task ϵ, process $\{(\mathcal{H}_\epsilon(t), \mathcal{X}(t))\}_{0\leq t\leq\theta_\epsilon}$ describes where copies of ϵ are stored, through which we define the device effort to quantify the storage resource consumption, as the impact (at the individual device level) of ϵ's movements on the nodes in set E.

Definition 4.2. *The* ***device effort*** *spent on task ϵ at time $t \in [t_\epsilon, t_\epsilon + \theta_\epsilon]$ is defined as the amount of local storage resource consumption (cumulative) in the carrying nodes,*

$$DE_\epsilon(t) := \int_{t_\epsilon}^{t} L_\epsilon|\mathcal{H}_\epsilon(s)| + \alpha_\epsilon L_\epsilon|\mathcal{H}_{\alpha(\epsilon)}(s)|\; ds, \tag{4.5}$$

where $|\mathcal{H}_\epsilon(s)|$ is the number of copies of task ϵ stored on any device/node in the network. Denote $DE_\epsilon := DE_\epsilon(t_\epsilon + \theta_\epsilon)$ as the total device effort spent during the offloading process of task ϵ.

Note that $|\mathcal{H}_\epsilon(t)| = ||\vec{\epsilon}(t)||_1$ is a random variable in probability space $(\Omega, \mathcal{F}, \mathbb{P})$, that takes range in $[0, |E|]$. The device effort quantifies the amount of storage resource consumed upto time t. For instance, during the offloading process shown in Figure 4.2, the amount of device effort spent on task ϵ are $2L_\epsilon$ during the queuing and processing stage, while it is L_ϵ during the offloading stage, and αL_ϵ during the responding stage. This is because both the task node k and the fog node n need to keep a copy of task ϵ to avoid transmission failure or processing failure, during the queuing and processing stage.

On the other hand, the *network effort* quantifies the amount of shared resource it takes the fog system, to perpetuate the movements of task ϵ, as a data block, over the air interface. By network resource, we mean communication resource belonging to the provisioning wireless network, *e.g.*, spectrum, resource blocks, transmission opportunities, and so on.

Definition 4.3. *For task ϵ with lifetime θ_ϵ, the* ***network effort*** *of ϵ at time $t \in [t_\epsilon, t_\epsilon + \theta_\epsilon]$ is defined as the sum of every distance-transmission time product, that is,*

$$NE_\epsilon(t) := \sum_{s \in \mathcal{T}^\epsilon} \sum_{i \in \Delta\mathcal{H}_\epsilon(t)} T^\epsilon_{p(i)\to i} d\Big(X_i(s), X_{p(i)}(s)\Big), \tag{4.6}$$

where $\Delta\mathcal{H}_\epsilon(t) := \mathcal{H}_\epsilon(t) \setminus \mathcal{H}_\epsilon(t^-)$ denotes the set of new recipients of data block ϵ at time t, set $\mathcal{T}^\epsilon := \{s \in (t_\epsilon, t] \mid \Delta\mathcal{H}_\epsilon(t) \neq \phi\}$ denotes all the time instants when data block ϵ is transmitted over the air, and $p(i)$ denotes the node, from which i receives a copy of ϵ. Let $NE_\epsilon := NE_\epsilon(t_\epsilon + \theta_\epsilon)$ denote the total network effort spent during the entire offloading process of task ϵ.

Note that the two types of efforts, *i.e.*, device effort and network effort, can not be directed added, because they quantify resource consumption in different types (and are hence in different units), but they jointly illustrate the amount of resources spent on the dissemination process of data blocks, and can be separately compared among offloading processes (and more generally, any data delivery process in a wireless network) of different tasks. Formulating the resource consumption of an individual offloading process as functions of paths taken by the data block, we are able to decouple the performance metrics from the offloading schemes. As a result, we have the expected answers to the first two research questions, irrespective of the criteria and implementation details of offloading algorithms.

4.3 How Data Move: the Gravity Model for Task Offloading

As identified in the moving object analogy, the offloading scheme functions as an external force to determine the moving direction of tasks. It seems straight-forward, but modeling such a mechanism is actually very challenging. First and foremost, the task node may use a variety of offloading schemes, which results in unmanageable space of network components for modeling. Second, offloading traffic increases as the fog system scales, so multiple nodes in the fog system may compete for shared resources, including storage, processing, and communication. Consequently, the offloading decision of one task node is inevitably affected by others. Lastly, for the cooperative offloading schemes, there is a probability that a task may not be

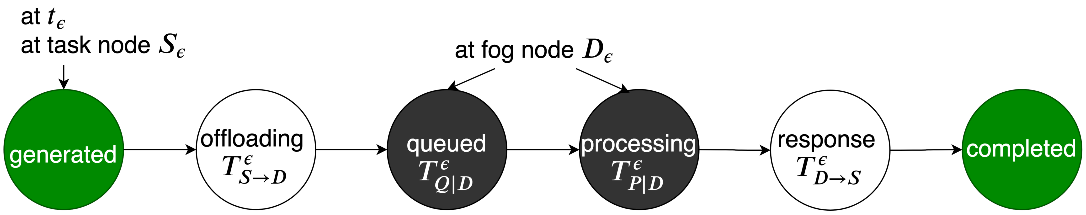

at t_ϵ
at task node S_ϵ
at fog node D_ϵ
generated
offloading $T^\epsilon_{S\to D}$
queued $T^\epsilon_{Q|D}$
processing $T^\epsilon_{P|D}$
response $T^\epsilon_{D\to S}$
completed

Figure 4.3 An offloading process from the perspective of task ϵ. Note that S_ϵ can be the same as D_ϵ for local processing, in which case transmission time $T^\epsilon_{S\to D} = T^\epsilon_{D\to S} = 0$.

able to be offloaded to the 'optimal' fog node seen from the task node's perspective, because of load-balancing operations among fog nodes.

To overcome these challenges, we propose a generic *gravity*-based offloading model, in which different offloading criteria (to select the target fog node) can be written as different forms of gravity functions, while allowing the task node to select any qualified candidates with a probability that is proportionate to its 'suitableness'. Based on this model, we are able to derive upper bounds of efforts spent on the offloading process of a single task. In this section, we focus on the simplest, yet most essential form of task offloading processes, that is, single-hop offloading, in which task ϵ moves from its generator, *i.e.*, task node $S_\epsilon \in E$, directly to a processor, *e.g.*, fog node $D_\epsilon \in E$.

4.3.1 An Offloading Procedure

The state transitions of task ϵ during its offloading process are shown in Figure 4.3. And the offloading process, as shown in Figure 4.2, can be described as follows:

1. At time t_ϵ, node $S_\epsilon = k \in E$ generates task ϵ, and $\mathcal{H}_\epsilon(t_\epsilon) = \{k\}$.

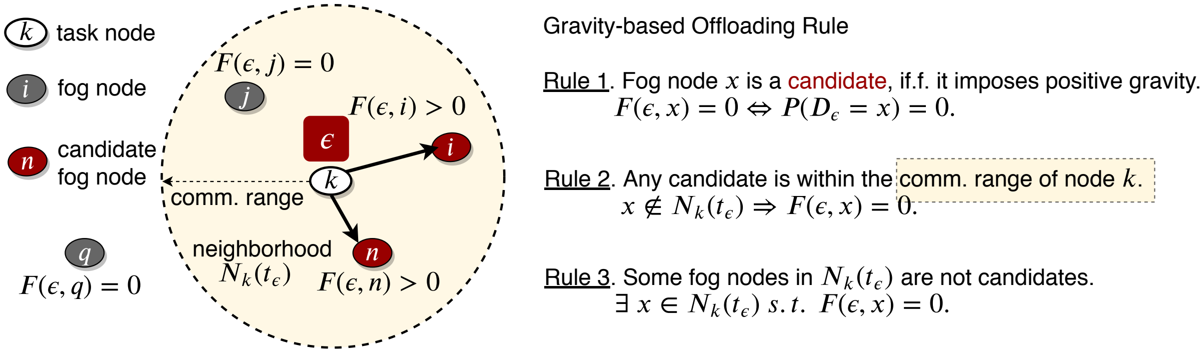
2. Task node k chooses the offloading target $D_\epsilon = n \in N_k(t_\epsilon) \cup \{k\}$ with probability $\mathbb{P}(D_\epsilon = n)$.

3. Task ϵ is then transmitted from node k to node n at data rate $R_{k\to n}$. As a result, $\mathcal{H}_\epsilon(t_\epsilon + T^\epsilon_{k\to n}) = \{n, k\}$. If $n = k$, *i.e.*, task node k made a decision to process task ϵ locally, this step is omitted, and $T^\epsilon_{k\to n} = 0$.

4. Upon receiving task ϵ, node n will put ϵ in a queue if processor of node n is busy (occupied by tasks from other offloading processes). Processing of ϵ will start once there is spare resource at n.

5. When processing of ϵ is completed (after queuing delay $T^\epsilon_{Q|n}$ and processing delay $T^\epsilon_{P|n}$), the corresponding result $\alpha(\epsilon)$, *e.g.* trans-coded video, object recognition output, or acknowledgement of a file uploading (to remote cloud servers), of size $\alpha_\epsilon L_\epsilon$, will be transmitted back to the task node k at the data rate of $R_{n\to k}$.

6. Finally, at time $t_\epsilon + T^\epsilon_{k\to n} + T^\epsilon_{Q|n} + T^\epsilon_{P|n} + T^\epsilon_{n\to k}$, the response $\alpha(\epsilon)$ is returned to node k.



Gravity-based Offloading Rule

Rule 1. Fog node x is a candidate, if.f. it imposes positive gravity.
$F(\epsilon, x) = 0 \Leftrightarrow P(D_\epsilon = x) = 0.$

Rule 2. Any candidate is within the comm. range of node k.
$x \notin N_k(t_\epsilon) \Rightarrow F(\epsilon, x) = 0.$

Rule 3. Some fog nodes in $N_k(t_\epsilon)$ are not candidates.
$\exists\, x \in N_k(t_\epsilon)\ s.t.\ F(\epsilon, x) = 0.$

Figure 4.4 An example of the gravity-based data offloading: every fog node $i \in N_k(t_\epsilon) \cup \{k\}$ (including the task node k itself) imposes a non-negative gravity force $F(\epsilon, i)$ on task ϵ, and the offloading probability from k to i is proportionate to the gravity force $F(\epsilon, i)$ defined in Eq. (4.7).

The first important feature of the proposed gravity model is allowing probabilistic target selection for load-balancing purposes. The offloading probability is defined as follows.

Definition 4.4. *The **offloading (to n) probability** $\mathbb{P}(D_\epsilon = n)$, that is, probability that fog node $n \in E$ is chosen by task node S_ϵ as the offloading target for task ϵ, is defined as*

$$\mathbb{P}(D_\epsilon = n) := \begin{cases} 0, & \textit{if } \sum_{i \in N_k(t_\epsilon)} F(\epsilon, i) = 0, \\ \frac{\rho(F(\epsilon,n))}{\sum_{i \in N_k(t_\epsilon)} \rho(F(\epsilon,i))}, & \textit{otherwise}, \end{cases} \tag{4.7}$$

where $F(\epsilon, i) \geq 0$ is the gravity on task ϵ imposed by fog node $i \in E$, and selectiveness function $\rho(\cdot) \geq 0$ is an increasing, convex function.

Note that when none of the candidate fog nodes imposes a positive gravity (*e.g.*, the transmission time alone is longer than the timeout value τ_ϵ) on task ϵ, *i.e.*, the first case in Eq. (4.7), task ϵ will be processed locally, *i.e.*, $D_\epsilon = S_\epsilon = k$. The probabilistic setting addresses the identified challenges in offloading modeling from the following aspects:

(1)*Diverse offloading criteria.* The definition of offloading probability provides two options with respect to modeling different criteria. First, gravity function $F(\epsilon, i)$ can be set by the task node to adapt to its unique requirements. Second, selectiveness function $\rho(\cdot)$ allows the task node to fine-tune how selective it is, or how reluctant it is to adjustments due to load-balancing. For example, we can set $\rho(x) = x^\gamma$, where $\gamma \in [1, 10]$ is the selectiveness coefficient that controls the extremeness of solution. The larger γ is, the more likely task node S_ϵ will favor fog nodes with the highest gravity, *i.e.*, probability of being selected are more concentrated on the top-ranking fog nodes.

(2) *Competition and adjustment.* The probabilistic setting achieves an 'automatic' load-balancing among fog nodes, in the sense that tasks are not always offloaded to the seemingly 'optimal' choice (fog node imposing the largest gravity force) from the (possibly incomplete) view of the task node, to avoid a small number of powerful fog nodes being overloaded by

excessive task offloading. This measure (allowing tasks to offloaded to sub-optimal fog nodes) can also be viewed as a result of the competition among task nodes.

4.3.2 Typical and Generic Gravity Rules

The gravity force $F(\epsilon, n)$ is meant to quantify the 'attractiveness' of fog node n, in regard to the processing of task ϵ, which can be freely defined by any task node. Considering most offloading criteria have an objective to optimize, such as delay, energy consumption, etc., we define the gravity function as the performance gain achieved by offloading, compared to local processing, or the remaining allowance against a performance budget. We list a few example rules for typical application scenarios.

4.3.2.1 Gravity Rule based on Distance

The simplest offloading scheme is to always offload to the nearest fog node, as long as there is one in the communication range of the task node, regardless of status of the fog nodes. This simple rule applies to offloading processes from IoT devices without local processing power, in which energy spent on data transmission over the air interface is the major concern, hence the preference on short distances. The gravity force for distance-based offloading can be written as follows:

$$F(\epsilon, n) = \frac{1}{d\left(X_k(t_\epsilon), X_n(t_\epsilon)\right)}, \ \forall n \in N_e(t_\epsilon). \tag{4.8}$$

To always select the nearest fog node, the probability mapping function $\rho(x)$ can be set to $\rho(x) = x^\gamma$ with a large γ, because the larger the selectiveness coefficient γ is, the more likely the task node chooses the nearest fog node.

4.3.2.2 Gravity Rule based on Energy Consumption

For pocketable, battery-operated task nodes, *e.g.*, IoT sensors, energy consumption for data processing is always the major concern. So the ultimate goal of offloading is to spend as less energy (mainly consumed by data transmission) as possible. In this sense, the gravity force imposed by node n upon task ϵ can be defined as

$$F(\epsilon, n) = \max\left\{E_k^\epsilon - \frac{P_e L_\epsilon}{R_{max} - \psi d\left(X_k(t_\epsilon), X_n(t_\epsilon)\right)}, \ 0\right\}, \tag{4.9}$$

where R_{max}, P_e, ψ are defined in (4.3), and E_k^ϵ denotes the local energy consumption (for data processing), if task ϵ is processed at node k, which is proportionate to $\phi_\epsilon L_\epsilon$. The max operator is employed to indicate that offloading is only an option, when it consumes less energy compared to local processing.

4.3.2.3 Gravity Rule based on Capacity

For computationally expensive tasks, *e.g.*, video trans-coding, the major concern of the task node S_ϵ is the capacity of candidate fog nodes, such that the application can run smoothly.

$$F(\epsilon, n) = \max\left\{\tau_\epsilon - \frac{L_\epsilon \phi_\epsilon}{c_n},\ 0\right\},\ \forall n \in N_e(t_\epsilon), \tag{4.10}$$

where c_n (cycles/s) is the capacity of candidate n. Subtracting the processing workload of task ϵ, the gravity force measures the capacity/computation margin after taking in ϵ, so the larger the capacity margin, the more likely that once offloaded, task ϵ will be processed with a shorter queuing delay. Here, capacity can refer to the processing power of fog nodes. However, it can be generalized to the bundled capability of service provisioning, *e.g.*, number of reserved channels (logically, TCP connections) to the cloud server, from which the processing/service time has the same format as the processing power, and hence the same definition in Eq. (4.10).

4.3.2.4 Gravity Rule based on Delay

As the primary motivation of fog paradigm [119], reducing service latency is the top design priority for many task offloading schemes. As shown in the delay decomposition in Figure 4.2 and Figure 4.3, for a task ϵ generated by $S_\epsilon = k$, three components of the total delay can be determined, including the uplink transmission delay $T^\epsilon_{S\to D}$, processing delay $T^\epsilon_{P|D}$, and downlink transmission delay $T^\epsilon_{D\to S}$, if the offloading decision is made as $D_\epsilon = n$:

$$\begin{cases} T^\epsilon_{k\to n} & = \frac{L_\epsilon}{R_{max} - \psi d(X_k(t_\epsilon), X_n(t_\epsilon))}, \\ T^\epsilon_{n\to k} & = \frac{\alpha_\epsilon L_\epsilon}{R_{max} - \psi d(X_k(t_\epsilon), X_n(t_\epsilon))}, \\ T^\epsilon_{P|n} & = \frac{L_\epsilon \phi_\epsilon}{c_n}. \end{cases} \tag{4.11}$$

The queuing delay $T^\epsilon_{Q|n}$, on the other hand, is a random variable, whose value is dependent on the offloading processes of *other* tasks. For instance, as shown in Figure 4.2, though task ϵ is offloaded to node n, it has to wait until node n has completed processing task δ, which is offloaded earlier. In fact, the queuing delay is bounded by $0 \le T^\epsilon_{Q|n} < \frac{Q_n(t_\epsilon)+Y_n}{c_n}$, where random variable $Y_n \in [0, \infty)$ is the workload offloaded to node n, during the transmission delay of task ϵ, *i.e.*, interval $[t_\epsilon, t_\epsilon + T^\epsilon_{k\to n}]$. Note that task node k, as a fog node, has also been accepting tasks offloaded to itself, hence the 'local processing' option of ϵ may also induce queuing delay, so we define the gravity force imposed on task ϵ, by a candidate fog node $n \in N_k(t_\epsilon)$, as

$$F(\epsilon, n) = \max\left\{\tau_\epsilon - \left(T^\epsilon_{k\to n} + T^\epsilon_{P|n} + T^\epsilon_{n\to k}\right),\ 0\right\}, \tag{4.12}$$

to avoid including the undetermined queuing time in the offloading criterion.

Next, we present a generic form of gravity force, whose parameters can be tuned to describe the four rules enumerated above, for the ease of analysis.

Definition 4.5. *The **gravity force** $F(\epsilon, n)$ imposed by fog node n on task ϵ is defined as*

$$F(\epsilon, n) = f\left(A_\epsilon - \frac{B_\epsilon}{c_n} + \frac{G_\epsilon}{g\left(d(X_k(t_\epsilon), X_n(t_\epsilon))\right)}\right), \tag{4.13}$$

quantities in which have the following properties:

- *Function $f : \mathbb{R} \to \mathbb{R}$ is non-decreasing, and for any $a > 0$, $f(a) > 0$ and $f(-a) = 0$, $f(0) = 0$. The simplest form of f is $f(x) = \max\{x, 0\}$.*
- *Function $g : \mathbb{R}^+ \to \mathbb{R}$ is a linear non-decreasing function of the distance $d(X_k(t_\epsilon), X_n(t_\epsilon))$.*
- *Quantities A_ϵ, B_ϵ, and G_ϵ are non-negative constants, which are only relevant to the attributes of task ϵ itself, and satisfy $B_\epsilon + G_\epsilon > 0$.*

The rationale of this function form in Definition 4.5 is that, despite their differences, the gist of all offloading rules is to find a fog node with the highest capacity, the shortest processing delay, and the least communication cost in terms of energy (*i.e.*, that is located near the task node's current position). Together with the selectiveness function $\rho(\cdot)$ in Eq. (4.7), the generic rule in Eq. (4.13) can describe the aforementioned gravity rules in this subsection. For instance, in the distance-based offloading scheme[3] (Eq. (4.8)), $A_\epsilon = B_\epsilon = 0$, $G_\epsilon = 1$, and function $g(x) = x$, while in the delay-based offloading scheme (Eq. (4.12)), $A_\epsilon = \tau_\epsilon$, $B_\epsilon = \phi_\epsilon L_\epsilon$, $G_\epsilon = (1 + \alpha_\epsilon)L_\epsilon$, and $g(x) = \psi x - R_{max}$. Note that the physical meaning of function $|g(x)|$ in Eq. (4.13) is the data transmission rate at distance x for delay or energy-based offloading.

Remark 4.3. *Note in Eq.* (4.8) *to* (4.12)*, the physical meaning of gravity force $F(\epsilon, n)$ is different, and is hence measured in different units. When a joint criterion is considered, e.g., a task is to be offloaded to a fog node that simultaneously satisfy an energy consumption goal and a processing delay goal, the new gravity force can be defined as a linear combination of Eq.* (4.8) *to* (4.12)*, with predetermined weights on each term to capture the preference of the task node. The resulting joint gravity function will have the same generic form as defined in Eq.* (4.13)*.*

With this generic function form of the gravity force, we next derive the probability for local processing and offloading, as a preparation toward obtaining the three performance metrics.

4.3.3 Offloading Options and Probabilities

Fog systems are expected to include a large number of devices (fog/task nodes), so it is highly probable that multiple offloading candidates co-exist for a task ϵ, in the vicinity of task node S_ϵ. Consequently, the actual efforts of the offloading process depend on whether to offload, and

[3] This assumption applies to a rather homogeneous setting, when the fog nodes and task nodes have similar hardware configurations. However, Eq. (4.13) also applies to the case of $R_n > R_k$, which is very common if candidate fog nodes are only located at APs or eNodeBs, because term $T^\epsilon_{n\to k}$ will be fairly small, especially when α_ϵ is small as well.

if so, which fog node to choose. To address this issue, we first discuss the offloading options and their probabilities for a given task ϵ.

Consider $\epsilon = (t_\epsilon, L_\epsilon, \tau_\epsilon, S_\epsilon, D_\epsilon, \phi_\epsilon, \alpha_\epsilon)$, *i.e.*, a task[4] of size L_ϵ, generated at t_ϵ by $S_\epsilon = k$, and with processing intensity ϕ_ϵ and volume change coefficient α_ϵ. For any fog node $n \in E$, let random variable $V_n = d(X_k(t_\epsilon), X_n(t_\epsilon))$ denote the distance between n and k, with probability density function (PDF) $f_V(v)$ ($v \in [0, +\infty)$), and random variable C_n denote its processing capacity, with PDF $f_C(c)$ ($c \in [c_{min}, c_{max}]$). First, we present the probability for local processing, that is, task ϵ will not be offloaded to any candidates in node k's neighborhood.

Lemma 4.1. *The probability that task node $S_\epsilon = k$ decides at time t_ϵ to process task ϵ locally is given by*

$$\mathbb{P}(D_\epsilon = k) = 1 - (1 - \mathbb{P}_{pos})^{N-1}. \tag{4.14}$$

For scenario when $A_\epsilon \neq 0$, and function $g(x) = \psi x - R_{max}$, probability $\mathbb{P}_{pos}$ can be calculated as

$$\mathbb{P}_{pos} = \int_{\frac{B_\epsilon}{A_\epsilon}}^{c_{max}} \int_0^{g^{-1}\left(\frac{cG_\epsilon}{B_\epsilon - cA_\epsilon}\right)} f_V(v) dv f_C(c) dc, \tag{4.15}$$

where c_{max} is the maximum capacity among fog nodes in E.

Proof. Task ϵ is forced to be processed locally at task node $S_\epsilon = k$, if and only if there is no proper candidate (fog) nodes from $N_k(t_\epsilon)$, or equivalently, none of the $N-1$ fog nodes in set E impose a positive gravity on ϵ. This explains the form of Eq. (4.14) with respect to $\mathbb{P}_{pos}$, which is the probability that a randomly chosen fog node n from $E \setminus \{k\}$ qualifies as an offloading candidate, that is, fog node n imposes a positive gravity force, $F(\epsilon, n) > 0$.

For any fog node $n \in E \setminus \{k\}$ to be considered as an offloading candidate, two conditions have to hold simultaneously: first, node n must reside in k's communication range, *i.e.*, the data transmission rate $R_{k \to n}(t_\epsilon) > 0$, which requires function $g(V_n) < 0$; second, the gravity force $F(\epsilon, n) > 0$ is strictly positive, which means there must be a performance gain if task ϵ is offloaded. Then these two requirements translate to the following sufficient condition:

$$\begin{cases} C_n & > \frac{B_\epsilon}{A_\epsilon}, \qquad \text{Condition 1,} \\ g(V_n) & < \frac{G_\epsilon}{\frac{B_\epsilon}{C_n} - A_\epsilon}, \qquad \text{Condition 2.} \end{cases} \tag{4.16}$$

As can be seen from Eq. (4.16), each capacity value $c \in [c_{min}, c_{max}]$ corresponds to a distance $v_{max} = g^{-1}\left(\frac{cG_\epsilon}{B_\epsilon - cA_\epsilon}\right)$, which explains the integral limits in Eq. (4.15). Note that the threshold distance v_{max} is smaller than or equal to the communication range, because when the Condition 1 on the first line of Eq. (4.16) holds, the right-hand-side of Condition 2 is guaranteed to be negative, so the communication range requirement $g(V_n) < 0$ will be satisfied. □

Lemma 4.1 provides a criterion of selecting offloading candidates from $N_k(t_\epsilon)$ for task ϵ, as

[4] For simplicity reasons, we assume $\tau_\epsilon = \frac{L_\epsilon \phi_\epsilon}{c_{S_\epsilon}}$.

well as the probability that ϵ will be processed locally at its generator k. It is a generic result, in the sense that for different scenarios, *e.g.*, pedestrians walking in a region under a random walk model, or vehicles moving along a highway, this lemma gives numeric results, as long as the distributions of node distance V_n and processing capacity C_n are known. It is also worth noticing that the sum of offloading probabilities (equivalently $1 - \mathbb{P}(D_\epsilon = k)$) can be obtained with statistics of the node ensemble, instead of exact parameters of each individual node. This indicates that, with the proposed gravity model, it is possible for a network operator to estimate the offloading performance though analysis, instead of lengthy simulation or costly experiments. Next, we discuss the offloading probability $\mathbb{P}(D_\epsilon = n)$ to a specific fog node n, whose capacity c_n and distance $d(X_n(t_\epsilon), X_k(t_\epsilon)) \overset{\triangle}{=} v_n$ are known by task node k at time t_ϵ.

Theorem 4.1. *Let $\mu_C = \int_{c_{min}}^{c_{max}} c f_C(c) dc$ denote the mean capacity of fog nodes in E, and $F_V(v) = \int_0^v f_V(s) ds$ denote the CDF of distance V between a randomly chosen fog node and task node k at time t_ϵ. The probability that task ϵ will be offloaded to a fog node n with capacity c_n and distance v_n, i.e., $\mathbb{P}(D_\epsilon = n)$, is upper bounded by*

$$\mathbb{P}(D_\epsilon = n) \leq \frac{\rho\left(F(\epsilon, n)\right)}{\mathrm{LB}(v_{max})\rho\left(A_\epsilon - \frac{B_\epsilon}{\mu_C} + \frac{G_\epsilon}{F_V(v_{max})}\int_0^{v_{max}} \frac{f_V(v)}{g(v)} dv\right)}, \tag{4.17}$$

where $v_{max} = g^{-1}(\frac{c_{max} G_\epsilon}{B_\epsilon - c_{max} A_\epsilon})$ is the maximum possible distance between a candidate node and task ϵ at time t_ϵ, and $\mathrm{LB}(v_{max}) = N F_V(v_{max}) f_C(c_{max}) \frac{-B_\epsilon G_\epsilon g(v_{max})}{(A_\epsilon g(v_{max}) + G_\epsilon)^2}$ is an lower bound of the total number of offloading candidates.

Proof. The key to derive the offloading (to n) probability $\mathbb{P}(D_\epsilon = n)$ in Eq. (4.7), is finding the denominator $\sum_{i \in N_k(t_\epsilon)} \rho\left(F(\epsilon, i)\right)$ in the definition.

First, as $\rho(\cdot)$ is a convex function, *e.g.*, $\rho(x) = x^\beta$, we have

$$\sum_{i \in N_k(t_\epsilon)} \rho\left(F(\epsilon, i)\right) \overset{Jensen}{\geq} |N_+(t_\epsilon)| \rho\left(\frac{\sum_{i \in N_+(t_\epsilon)} F(\epsilon, i)}{|N_+(t_\epsilon)|}\right), \tag{4.18}$$

where $N_+(t_\epsilon \subset N_k(t_\epsilon$ is the set of candidate fog nodes (imposing a positive gravity force on task ϵ).

The next step is to obtain $|N_+(t_\epsilon)|$ and $\sum_{i \in N_k(t_\epsilon)} F(\epsilon, i)$, that is, the total number of candidate nodes, and the total gravity force imposed by all candidates on task ϵ.

Observing that the gravity force only concerns the capacity C_i and distance V_i of a node i, and both quantities are additive, we resort to a divide-and-conquer method for the solution of $|N_+(t_\epsilon)|$ and $\sum_{i \in N_+(t_\epsilon)} F(\epsilon, i)$. To be more specific, we divide the disk area (yellow shades) centered at $X_k(t_\epsilon)$ into circular regions of width Δv, as shown in Figure 4.5.

Consider the circular region with inner radius $v \in [0, v_{\max}]$, as indicated by the blue ring with width Δv. When Δv is small, the probability that a node falls into this region can be approximated as $f_V(v)\Delta v$. The probability that an arbitrary fog node i is located inside this

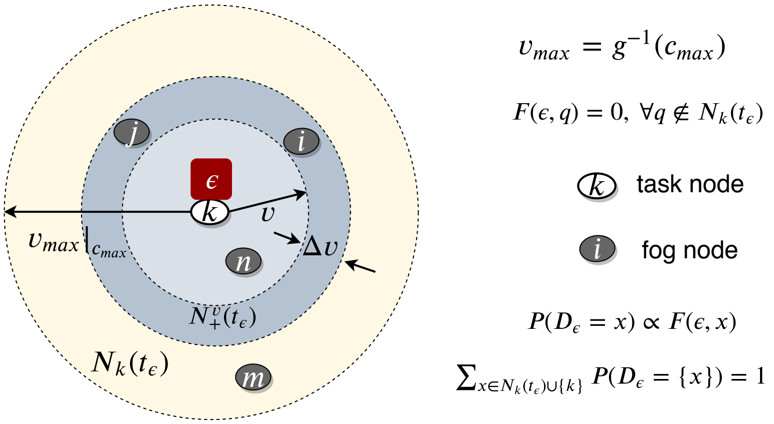

ϵ
j
i
k
v
$v_{max}|_{c_{max}}$
Δv
n
$N_+^v(t_\epsilon)$
$N_k(t_\epsilon)$
m
$v_{max} = g^{-1}(c_{max})$
$F(\epsilon, q) = 0, \ \forall q \notin N_k(t_\epsilon)$
k task node
i fog node
$P(D_\epsilon = x) \propto F(\epsilon, x)$
$\sum_{x \in N_k(t_\epsilon) \cup \{k\}} P(D_\epsilon = \{x\}) = 1$

Figure 4.5 Proof technique in Theorem 4.1: dividing the neighborhood of task node k into multiple rings of width Δv.

circular region, and satisfies the criterion in (4.16), can be calculated as

$$p_0(v) = f_V(v)\Delta v \int_{\frac{B_\epsilon g(v)}{A_\epsilon g(v)+G_\epsilon}}^{c_{max}} f_C(c)dc = f_V(v)\Delta v \left[1 - F_C\left(\frac{B_\epsilon g(v)}{A_\epsilon g(v) + G_\epsilon}\right)\right], \tag{4.19}$$

where $F_C(c)$ is the CDF of random variable C, representing the capacity of a randomly chosen fog node from set E.

Denote the set of candidate fog nodes (imposing a positive gravity force on task ϵ) inside the blue ring as $N_+^v(t_\epsilon)$. It is easy to see that random variable $|N_+^v(t_\epsilon)| \sim \mathbf{B}(N, p_0(v))$, with mean

$$\mathbb{E}(|N_+^v(t_\epsilon)|) = N f_V(v)\Delta v \left[1 - F_C\left(\frac{B_\epsilon g(v)}{A_\epsilon g(v) + G_\epsilon}\right)\right]. \tag{4.20}$$

Then the total number of candidates in the yellow disk $\mathbb{E}\left(|N_+(t_\epsilon)|\right)$ can be lower bounded as

$$\begin{aligned}\mathbb{E}\left(|N_+(t_\epsilon)|\right) &= \mathbb{E}\left(\int_0^{v_{max}} |N_+^v(t_\epsilon)| dv\right) \overset{Fubini}{=} \int_0^{v_{max}} \mathbb{E}\left(|N_+^v(t_\epsilon)|\right) dv \\ &\geq N F_V(v_{max}) f_C(c_{max}) \frac{-B_\epsilon G_\epsilon g(v_{max})}{(A_\epsilon g(v_{max}) + G_\epsilon)^2},\end{aligned} \tag{4.21}$$

where $g(v_{max}) < 0$ is guaranteed by the candidate criterion in Eq. (4.16), such that $\mathbb{E}\left(|N_+(t_\epsilon)|\right)$, is always positive. Meanwhile, the expected number of candidates $\mathbb{E}\left(|N_+(t_\epsilon)|\right)$ is also upper-bounded by the total number of nodes within the yellow disk, which also contains nodes that fail criterion Eq. (4.16), that is,

$$\mathbb{E}\left(|N_+(t_\epsilon)|\right) \leq N F_V(v_{max}). \tag{4.22}$$

On the other hand, the total gravity force on task ϵ can be calculated as the sum of all Δv

rings within the disk of radius v_{max}, that is,

$$\begin{aligned}
&\mathbb{E}\left(\sum_{i\in N_k(t_\epsilon)} F(\epsilon,i)\right) \overset{Fubini}{=} \int_0^{v_{max}} \mathbb{E}\left(\sum_{i\in N_k^v(t_\epsilon)} F(\epsilon,i)\right)\\
&\geq N\int_0^{v_{max}} f_V(v)\left[1-F_C\left(\frac{B_\epsilon g(v)}{A_\epsilon g(v)+G_\epsilon}\right)\right]\int_{\frac{B_\epsilon g(v)}{A_\epsilon g(v)+G_\epsilon}}^{c_{max}}\left(A_\epsilon+\frac{G_\epsilon}{g(v)}-\frac{B_\epsilon}{c}\right)f_C(c)dcdv\\
&\geq N\left(A_\epsilon-\frac{B_\epsilon}{\mu_C}\right)F_V(v_{max})+NG_\epsilon\int_0^{v_{max}}\frac{f_V(v)}{g(v)}dv,
\end{aligned} \tag{4.23}$$

where $\mu_C=\mathbb{E}(C)$ is the mean capacity of fog nodes in set E.

Consequently, the probability that task ϵ starts to offload to node n is upper bounded by

$$\mathbb{P}(D_\epsilon=\{n\})\leq \frac{\rho\left(F(\epsilon,n)\right)}{|N_+(t_\epsilon)|\rho\left[\frac{\mathbb{E}\left(\sum_{i\in N_+(t_\epsilon)}F(\epsilon,i)\right)}{|N_+(t_\epsilon)|}\right]}, \tag{4.24}$$

plugging Eq. (4.23), upper bound Eq. (4.22), and lower bound Eq. (4.21) into the above equation yields the result in Eq. (4.17). □

We highlight two implications from the upper bound in Theorem 4.1:

(1) *Invariant property.* Regardless of the gravity rule and selectiveness, offloading probability $\mathbb{P}(D_\epsilon=n)=O(\frac{1}{N})$, that is, this probability is asymptotically bounded above by $\frac{1}{N}$, when the mapping function $\rho(\cdot)$ takes a polynomial form, *i.e.* $\rho(x)=x^\beta+x^{\beta-1}+\cdots+x+1$.

(2) *Local action and global status.* With this theorem, we establish a link between a local action (task node k chooses fog node n to offload task ϵ) to the global status of the system, because in this process, both device effort and network effort for task ϵ take changes, which in turn affects the offloading options, and hence the efforts spent for other tasks. In other words, with the set of probability $\{\mathbb{P}(D_\epsilon=n)\}_{n\in E}$, we can examine interactions of multiple tasks, and their movements after time instant t_ϵ in the fog system.

4.4 Bounds and Scaling Laws of Performances

Recall that, in the one-hop data offloading process, the lifetime, device effort and network effort of a single task ϵ can be obtained by their definitions once the offloading target D_ϵ is chosen to be fog node $n\in E$, that is,

$$\begin{cases}
\theta_\epsilon^n & = T_{k\to n}^\epsilon+T_{Q|n}^\epsilon+T_{P|n}^\epsilon+T_{n\to k}^\epsilon,\\
\mathrm{DE}_\epsilon^n(\theta_\epsilon) & = L_\epsilon\left(T_{k\to n}^\epsilon+2T_{Q|n}^\epsilon+2T_{P|n}^\epsilon\right)+\alpha_\epsilon L_\epsilon T_{n\to k}^\epsilon,\\
\mathrm{NE}_\epsilon^n(\theta_\epsilon) & = T_{k\to n}^\epsilon d(X_k(t_\epsilon),X_n(t_\epsilon))+T_{n\to k}^\epsilon d(X_k(\theta_\epsilon+t_\epsilon),X_n(\theta_\epsilon+t_\epsilon)),
\end{cases} \tag{4.25}$$

as illustrated by the example shown in Figure 4.2.

The exact value of the three performance metrics on the offloading process of a task ϵ can only be evaluated, when the target fog node n is chosen. However, in a probabilistic sense, we can derive the expected lifetime and efforts for a given task ϵ, to understand the system-level behavior. In this case, every term in Eq. (4.25) is only dependent on parameters (c_n, V_n) of fog node n, and the task ϵ, except for the queuing delay $T^{\epsilon}_{Q|n}$, which is the result of multiple offloading processes in addition to task ϵ. Therefore, we first analyze the condition in which other offloading processes will affect the queuing delay $T^{\epsilon}_{Q|n}$ for task ϵ.

4.4.1 Discussion on Expected Queuing Delay $\mathbb{E}(T^{\epsilon}_{Q|n})$

Consider another task δ that is also offloaded to fog node n with probability $\mathbb{P}(D_\delta = n)$. Its processing may add to the queuing delay if and only if the last bit of δ arrived at n before the last bit of ϵ, such that δ will be processed before ϵ, that is, $t_\epsilon < t_\delta + T^{\delta}_{S_\delta \to n} < t_\epsilon + T^{\epsilon}_{S_\epsilon \to n}$.

According to the task node model described in Section 4.2, the data traffic generation process of every node in set E is stationary, *i.e.*, the distribution of intervals between two consecutive tasks generated by the same node does not change over time, and the mean interval length is $\frac{1}{\beta}$. In other words, the system will generate $N\beta$ tasks on average, creating $N\beta\mu_L\mu_\phi$ processing load, for each time step. If the total processing capacity $N\mu_C$ is smaller than the generated processing load, the expected queuing time will eventually tend to infinity, in which case it is less interesting to examine the performance of the fog system. So we consider the case when generated loaded can be processed, that is, $N\mu_C > N\beta\mu_L\mu_\phi$.

Assume the distributions of different parameters of the task, *e.g.*, data volume L_δ and processing intensity ϕ_δ, are independent of each other. Then with a similar divide-and-conquer method as depicted in Figure 4.5, we can upper bound the expected amount of data $\mathbb{E}(Y_n)$ offloaded to (arrived at) n during time interval $[t_\epsilon, t_\epsilon + T^{\epsilon}_{S_\epsilon \to n}]$ by considering all the tasks generated within fog node n's communication range $g^{-1}(0)$, that is,

$$\mathbb{E}(Y_n) \leq \beta N T^{\epsilon}_{k\to n} \int_0^{g^{-1}(0)} \frac{f_V(v)}{|N_+(t_\epsilon)|} \int_0^{T^{\epsilon}_{k\to n}} \mu_\phi s H_L \left[g(v)(s - T^{\epsilon}_{k\to n})\right] ds dv, \tag{4.26}$$

where μ_ϕ is the expected processing intensity of tasks, and function $H_L(l) = \int_0^l x f_L(x)dx$ is the partial expectation of data volume L, which is less than or equal to threshold l. Note that the denominator $|N_+(t_\epsilon)|$ follows from the upper bound of offloading probability in Eq. (4.24). Then, the expected queuing time is upper bounded by

$$\mathbb{E}(T^{\epsilon}_{Q|n}) \leq \frac{\mathbb{E}(Y_n)}{c_n} + \max\left\{0, \frac{Q_n(t_\epsilon)}{c_n} - T^{\epsilon}_{k\to n}\right\}, \tag{4.27}$$

where $Q_n(t_\epsilon)$ is the remaining load in fog node n at time t_ϵ.

From a realistic point of view, the peak data rates in existing LTE and WiFi systems measure at 10 Mbps to 100 Mbps, while the processing capacity of most mobile devices measure at GHz level (*e.g.*, 1.4 GHz Raspberry Pi 3, and 2.6∼3.0 GHz smart phone processors), so the second term of Eq. (4.27) is less significant compared to the first term. Considering that $|N_+(t_\epsilon)| = \Theta(N)$, as shown in the proof of Theorem 4.1, we have $\mathbb{E}(T^\epsilon_{Q|n}) = O(\frac{1}{N})$ from Eq. (4.26) and (4.27), especially when β is large. On the other hand, when β is small, that is, the system is lightly loaded, the probability that tasks arrives at the end of transmission will be small, such that $\mathbb{E}(T^\epsilon_{Q|n}) \simeq \max\left\{0, \frac{Y_n}{c_n} - T^\epsilon_{k\to n}\right\}$ will also be small positive values.

With Eq. (4.27) and (4.25), we are able to obtain the device and network efforts for ϵ, given that it is offloaded to node n. Consequently, an upper bound of the device effort DE_ϵ, can be obtained by $\mathbb{E}(\mathrm{DE}_\epsilon) = \sum_{i\in N_+(t_\epsilon)\cup\{k\}} \mathrm{DE}^n_\epsilon P(D_\epsilon = i)$, where the device effort DE_ϵ can be replace by lifetime θ_ϵ, or network effort NE_ϵ.

Despite the complex form of the expected queuing delay, it outlines a generic way to foresee task ϵ's resource consumption during the offloading process, irrespective to different initial condition and network status. In some special cases, *e.g.*, the distance V_n follows a special distribution, we are able to obtain concrete, and more accurate results, so in the remaining of this section, we consider the special case when fog nodes follow a random walk model.

Remark 4.4. *Without loss of generality, we assume the data volume L and processing intensity ϕ of tasks satisfy uniform distributions, with mean μ_L and μ_ϕ respectively. Consider the selectiveness function takes the form of $\rho(x) = x$. Consider nodes in E are initially randomly located in a square region $[-\frac{a}{2}, \frac{a}{2}]^2 \subset \mathbb{R}^2$, and they all move under a speed-constrained random walk model with a maximum speed $speed_{max} << a$. When a node hits the boarder of the square region, it shows up on the opposite boarder of the region, such that there is no boarder effect [11] in this scenario. In this case, the distance V between any two randomly chosen entities satisfies the following lemma.*

Lemma 4.2. *(Special case of [92].) Consider a square region $[-\frac{a}{2}, \frac{a}{2}]^2 \subset \mathbb{R}^2$. For two randomly chosen points $X_1, X_2 \in [-\frac{a}{2}, \frac{a}{2}]^2$ in this region, the distance $V = d(X_1, X_2)$ is a random variable with PDF as follows:*

$$f_V(v) = \begin{cases} \frac{\pi}{a^2} + \frac{v^2}{a^4}, & 0 \le v \le a, \\ -\frac{2+\pi}{a^2} + \frac{2}{a^2}\arcsin(\frac{a}{v}) + \frac{4(s^2-a^2)^{1/2}}{a^3}, & a \le v \le \sqrt{2}a. \end{cases} \tag{4.28}$$

Proof of Lemma 4.2 is very similar to the general case presented in [92], and is hence omitted here. In this scenario, we discuss data offloading processes under two gravity rules.

4.4.2 Performance under Distance-based Offloading

First, we discuss the simplest distance-based offloading rule, in which any task is for sure to be offloaded to a neighboring fog node, by its gravity function Eq. (4.8). For task ϵ under the this rule, the only criterion is transmission range, *i.e.*, $v_{max} = \frac{R_{max}}{\psi}$, because $F(\epsilon, n) > 0$ for any fog node $n \in E$.

Note that, by Lemma 4.2, the probability that two entities happen to be located at the exact same location $F_V(v = 0)$ is strictly zero, and that it is very unlikely (with probability less than $\frac{\pi}{a^2} + \frac{1}{3a^4}$) that their distance is less than 1 unit length, especially when a is large. Therefore, the gravity force $F(\epsilon, i)$ is well-defined ($< \infty$) for almost all offloading candidates. With this lemma, we derive the lifetime and efforts of an individual task ϵ in two steps. First, we give the offloading probability, device and network effort to a fixed candidate fog node n.

Proposition 4.1. *Consider a distance-based offloading scheme, with the selectiveness function $\rho(x) = x$. For task ϵ offloaded to fog node n, the efforts spent for task ϵ are given by*

$$\mathbb{E}(\mathrm{NE}_\epsilon^n) = \frac{(1+\alpha)L_\epsilon V_n}{R_{max} - \psi V_n}, \tag{4.29}$$

$$\mathbb{E}(\mathrm{DE}_\epsilon^n) = L_\epsilon \left[\frac{(1+\alpha)L_\epsilon}{R_{max} - \psi V_n} + \frac{2\phi_\epsilon L_\epsilon}{c_n} + 2\mathbb{E}(T_{Q|n}) \right], \tag{4.30}$$

where $\mathbb{E}(T_{Q|n}) \leq \frac{\beta \mu_\phi R_{max} L_\epsilon^3}{12 N c_n (R_{max} - \psi V_n)^3}$ is the expected queuing time at node n, and μ_ϕ is the mean processing intensity of all tasks.

Proof. Under the distance-based gravity field, the gravity rule can be stated as $F(\epsilon, n) = \frac{1}{V_n}$ for $V_n < v_{max}$, and $F(\epsilon, n) =$ otherwise, where $v_{max} = \frac{R_{max}}{\psi}$.

Plugging in the PDF $f_V(v)$ from Lemma 4.2, we obtain the expected number of offloading candidates with a similar method in Theorem 4.1, as $\mathbb{E}(|N_+(t_\epsilon)|) = N F_V(v_{max})$, and the offloading probability can be derived as

$$\mathbb{P}(D_\epsilon = n) \simeq \left(V_n N \int_1^{v_{max}} \frac{f_V(v)}{v} dv \right)^{-1} = \frac{2a^4}{N V_n \left[(v_{max})^2 + \pi a^2 \ln v_{max} \right]}, \tag{4.31}$$

Note the integral in the denominator of Eq. (4.31) starts from 1 to avoid a diverging sum, which is reasonable because the probability that the distance $V < 1$ is very small.

Then from Eq. (4.26), the expected amount of data $\mathbb{E}(Y_n)$ offloaded to entity n during the transmission time of task ϵ can be derived as

$$\begin{aligned} \mathbb{E}(Y_n)|_{T_{k\to n}^\epsilon} =& N \int_1^{v_{max}} f_V(v) \mathbb{P}(D_\epsilon = n) \beta s \mu_\phi \int_0^{T_{k\to n}^\epsilon} s H_L \left[(T_{k\to n}^\epsilon - s)(R_{max} - \psi v) \right] ds dv \\ =& \frac{\beta \mu_\phi R_{max}}{12N} \left(T_{k\to n}^\epsilon \right)^3, \end{aligned} \tag{4.32}$$

where $T_{k\to n}^\epsilon = \frac{L_\epsilon}{R_{max} - \psi V_n}$ is the transmission time of ϵ from the task node k to the candidate fog node n. Note that the partial expectation of data volume

$$H_L \left[(T_{k\to n}^\epsilon - s)(R_{max} - \psi v) \right] = \frac{1}{2} \left(L_{min} + (T_{k\to n}^\epsilon - s)(R_{max} - \psi v) \right), \tag{4.33}$$

because data volume L is uniformly distributed, and the lower bound L_{min} of data volume is small, so we can approximate this partial expectation with $\frac{1}{2}(T_{k\to n}^\epsilon - s)(R_{max} - \psi v)$.

Therefore, the expected queuing time $\mathbb{E}(T_{Q|n}) \leq \frac{\beta \mu_\phi R_{max} L_\epsilon}{12 N c_n} (T_{k\to n}^\epsilon)^3$. Plugging these quantities in Eq. (4.25) yields the network and device efforts. □

Proposition 4.1 gives the exact expected effort for task ϵ, if it is offloaded to fog node n, which happens with probability in Eq. (4.31). Then, we have the following corollary with respect to the expected effort spent on ϵ, in terms of network size N.

Corollary 4.1. *Under the distance-based offloading rule, the expected efforts to offload any task, i.e., the network effort $\mathbb{E}(NE_\epsilon)$ and the device effort $\mathbb{E}(DE_\epsilon)$, are $O(1)$ with respect to network size N.*

Proof. (Sketch.) The probability that task ϵ is offloaded to fog node n is shown in Eq. (4.31), which is $O(\frac{1}{N})$. For each node n, the expected effort for task ϵ, given that it is offloaded to n, are shown in Eq. (4.29) and (4.30), both of which are $O(1)$ with respect to N.

Applying the divide-and-conquer method shown in Figure 4.5, as that in the proof of Theorem 4.1, gives us the $O(1)$ scaling law, as $\mathbb{E}(|N_k(t_\epsilon)|) \sim \Theta(N)$, as shown by Eq. (4.21) and (4.22). For instance, the expected network effort for task ϵ is

$$\mathbb{E}(\mathrm{NE}_\epsilon) = \frac{2(1+\alpha)L_\epsilon \left[(a^2\pi + \frac{R_{max}}{\psi^2}) \ln R_{max} - \frac{3R_{max}^2}{2\psi^2} \right]}{\frac{R_{max}^2}{\psi} + \pi a^2 \psi \ln \frac{R_{max}}{\psi}}, \tag{4.34}$$

which is $O(1)$ with respect to N. □

Corollary 4.1 states that, the network effort and the device effort of a single task do not scale with the network size $N = |E|$. Further, if we consider all the tasks generated during the offloading process of ϵ as the total offloading traffic of the fog system, the total expected efforts, both DE and NE, scales as $O(N)$, due to the $O(N)$ number of new tasks generated per unit time. This indicates the linear growth of both traffic leads to a linear growth in resource demand, under the distance-based offloading scheme.

4.4.3 Performance under Delay-based Offloading

Next, we consider the most complicated offloading criterion, *i.e.*, delay-based offloading depicted by Eq. (4.12), because other gravity rules, *e.g.*, the power-based offloading rule in Eq. (4.9) and the capacity-based offloading rule in Eq. (4.10), can be viewed as specially cases of Eq. (4.12), in which two variables related to fog node n, $d(X_k(t_\epsilon), X_n(t_\epsilon))$ and C_n, reduce to one.

We adopt the same assumption as in the distance-based offloading: there are a total number of $N = |E|$ fog nodes in the region $[-\frac{a}{2}, \frac{a}{2}]^2$, in which a is large enough, such that $a >> \max_{e \in E}\{r_e\}$. Suppose a task node k located at the center of this region, *i.e.*, location $(0, 0)$, and generates task ϵ to offload at time t_ϵ. Denote the distance from fog node n to this task node k as $V_n^\epsilon := d(X_k(t_\epsilon), X_n(t_\epsilon))$, which obeys the distribution in Lemma 4.2. Under the delay-based offloading rule, we have the following corollary.

Corollary 4.2. *Under the delay-based offloading rule, the expected efforts to offload any task are also $O(1)$, with respect to the size of the fog network N.*

Proof. (Sketch.) Under the delay-based gravity rule, the criterion $F(\epsilon, n) > 0$ in Eq, (4.16) becomes

$$F(\epsilon, n) > 0 \Leftarrow \begin{cases} C_n & > \frac{\phi_\epsilon L_\epsilon}{\tau_\epsilon}, \\ V_n^\epsilon & < \frac{1}{\psi}\left[R_{max} - \frac{(1+\alpha_\epsilon)L_\epsilon}{\tau_\epsilon - \frac{\phi_\epsilon L_\epsilon}{C_n}}\right]. \end{cases} \tag{4.35}$$

From this criterion, especially the bound on distance V_n^ϵ, we can see that the possible offloading distance increases monotonically with the capacity of the fog node. Assume the capacity of any fog node is uniformly distributed on $[c_{min}, c_{max}]$. Then the maximum distance for the task node k to choose its offloading targets can be upper bounded by plugging $C_n = c_{max}$ into Eq. (4.35). In addition, the probability that a randomly chosen fog node n qualifies as a candidate can be calculated as

$$\mathbb{P}(F(\epsilon, n) > 0) = \int_{\frac{\phi_\epsilon L_\epsilon}{\tau_\epsilon}}^{c_{max}} \frac{\pi a^2 v + \frac{v^3}{3}}{a^4(c_{max} - c_{min})}\Bigg|_0^{\frac{1}{\psi}\left[R_{max} - \frac{(1+\alpha_\epsilon)L_\epsilon}{\tau_\epsilon - \frac{\phi_\epsilon L_\epsilon}{c}}\right]} dc. \tag{4.36}$$

Given that $N_k(t_\epsilon) \neq \phi$, we consider a particular fog node $n \in E$ with capacity $C_n = c_n$, which is located at a distance of $V_n^\epsilon = v_n$ from the task node k. Let

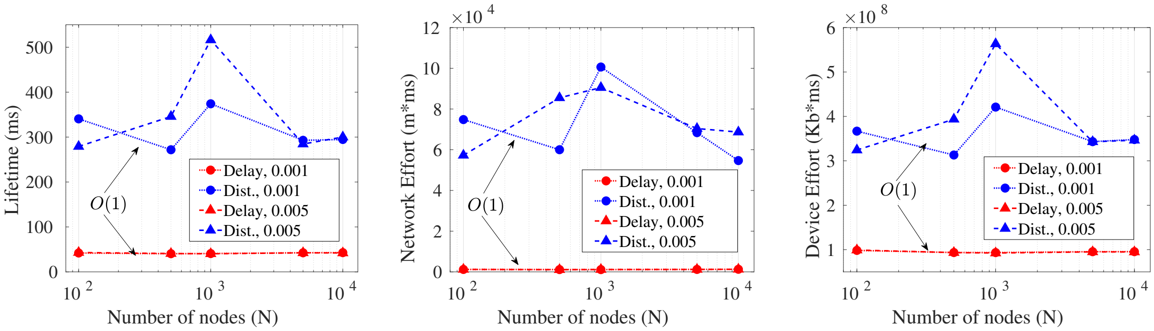
$$v_{\max} = \frac{1}{\psi}\left[R_{max} - \frac{(1+\alpha_\epsilon)L_\epsilon}{\tau_\epsilon - \frac{\phi_\epsilon L_\epsilon}{c_{max}}}\right] \tag{4.37}$$

denote the maximum possible distance in criterion (5.15) obtained by pushing C_n to its maximum value c_{max}. For an arbitrary fog node $q \in E$, when $V_q^\epsilon > v_{\max}$, *i.e.*, node q is located outside the yellow outer circle in Figure 4.5, the gravity force it imposes on task ϵ will be strictly zero, such that it will never be considered as a proper candidate, *i.e.*, $P(q \in D_\epsilon) = 0$. Therefore, we consider fog node n with $v_n \leq v_{max}$.

Similarly as in the proof of Theorem 4.1, we use the divide-and-conquer method to find the expected sum of gravity force on task ϵ, which is also the denominator in Eq. (4.7), as

$$\begin{aligned} \mathbb{E}\left(\sum_{i \in N_k(t_\epsilon)} F(\epsilon, i)\right) &= \int_{v=0}^{v_{max}} \mathbb{E}\left(\sum_{i \in N_k^v(t_\epsilon)} F(\epsilon, i)\right) \\ &= \int_0^{v_{max}} \frac{2\pi N v}{a^2}\left[\tau_\epsilon + \frac{(1+\alpha_\epsilon)L_\epsilon}{\psi v - R_{max}} - \frac{2\phi_\epsilon L_\epsilon}{c_{min} + c_{max}}\right] dv \\ &= \frac{\pi N}{a^2}\left[\tau_\epsilon v_{max}^2 + \frac{2(1+\alpha_\epsilon)L_\epsilon}{\psi} \ln\left(1 - \frac{\psi v_{max}}{R_{max}}\right) - \frac{4\phi_\epsilon L_\epsilon v_{max}}{c_{min} + c_{max}}\right]. \end{aligned} \tag{4.38}$$

From Eq. (4.38), we know that the probability that task ϵ is offloaded to fog node n is $O(\frac{1}{N})$, as the numerator in the probability definition is a constant $O(1)$ in terms of N. Then for each node n, the expected effort for task ϵ, given that the task is offloaded to n, are also $O(1)$ with respect to N. Applying the divide-and-conquer method shown in Figure 4.5, as that in the proof of Theorem 4.1, gives us the $O(1)$ scaling law, because $\mathbb{E}(|N_k(t_\epsilon)|) \sim \Theta(N)$ holds for the



(a) Mean lifetime $\mathbb{E}(\theta)$. **(b)** Mean network effort $\mathbb{E}(\mathrm{NE})$. **(c)** Mean device effort $\mathbb{E}(\mathrm{DE})$.

Figure 4.6 Numerical results show that the mean lifetime and efforts are $O(1)$ for a single task with respect to network size N, when the processing load is light ($\beta = 0.001$ and 0.005).

generic gravity rule, including delay-based offloading procedures. □

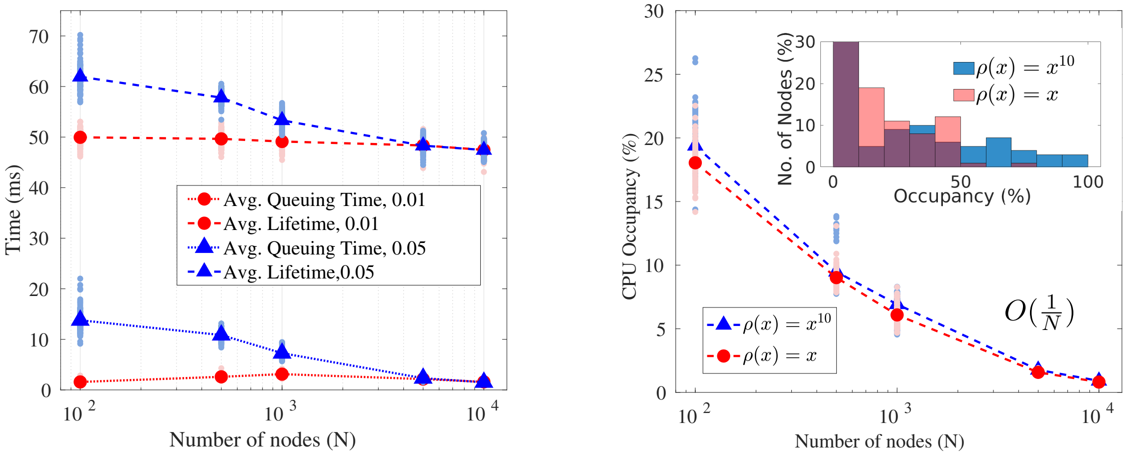
4.4.4 Numerical Results and Discussions

To validate the proposed gravity-based offloading model, and the derived scaling law of the three performance metrics, we simulate the offloading process for fog systems of different sizes. The simulation configuration is shown in Table 4.3. The mean lifetime, device and network efforts for the 100 (Figure 4.6) or 1000 (Figure 4.7-4.8) tasks generated during the simulation are shown in Figure 4.6-4.8. For each network size N, the simulation is run for 100 times, each of which has randomly changing initial configurations (results shown in light colored dots), to obtain the ensemble mean (results shown in bright colored markers and lines).

Table 4.3 Simulation configuration for Section 4.4.4.

Parameter	Description	Value/Range
$N = \lvert E \rvert$	network size	100, 500, 1000, 5000, 10000
a	region edge	1000 m
N/A	simulation run times	100
N/A	# of tasks (each simulation)	100 (Figure 4.6), 1000 (Figure 4.7-4.8)
β	task generation prob. (per ms)	0.001, 0.005, 0.01, 0.05
$speed_{max}$	max moving speed	10 m/s
ψ	data rate attenuation (see Eq. (4.3))	20 Mbps/m
R_{max}	max data rate	10 Gbps (considering [39, 109, 41, 52])
L	data volume	1Mb, 10 Mb, 100 Mb, 1 Gb
ϕ	processing intensity	100, 200, 300, 400 cycles/bit
α	data volume change coefficient	0.1, 0.5, 1
C	CPU frequency	1 GHz (w.p. 0.4), 4GHz (w.p. 0.6)

As can be seen from Figure 4.6, it is clear that despite the increase in network size N, the



(a) Lifetime and queuing time.

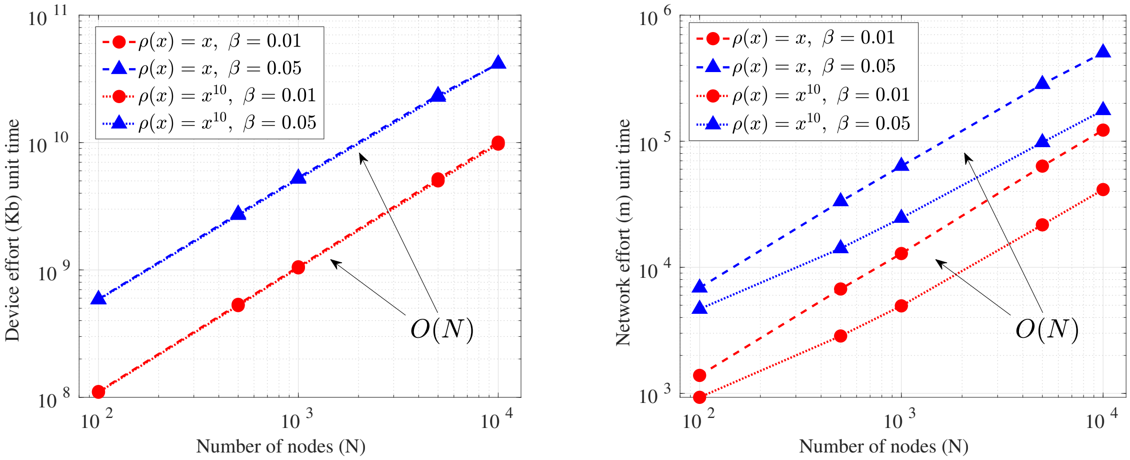
(b) CPU occupancy ($\beta = 0.05$) for different $\rho(x)$.

Figure 4.7 Under the delay-based offloading, lifetime of individual tasks and CPU occupancy at fog nodes are $O(\frac{1}{N})$, when the processing load is heavier ($\beta = 0.01$ and 0.05).

mean efforts remain constant, especially for the delay-based offloading scheme (flat red lines). For the distance-based offloading scheme (blue lines), the efforts are more volatile, because it pursues to offload every task, such that there a larger probability for a task to be offloaded to a less powerful fog node, which leads to unpredictable processing time, and hence the zig-zag pattern in these figures. Note that this result is obtained for offloading services within a fixed geographical region, in which the total amount of network resource is limited.

Next we examine the delay-based offloading scheme in the scenario of higher load $\beta = 0.01$ and $\beta = 0.05$. In Figure 4.7a, we observe the decrease in queuing time (dotted lines) and hence the lifetime (dashed lines) of individual tasks, as the network size grows. The decreasing trend is especially noticeable when the processing load is heavier, *i.e.* $\beta = 0.05$ (blue lines with triangle markers), as discussed in Section 4.4.1 for the expected queuing delay $\mathbb{E}(T_{Q|D})$ of tasks. This $O(\frac{1}{N})$ decreasing trend can be observed even more clearly by the amount of tasks offloaded ($\mathbb{E}(Y_n)$ in Eq. (4.26)) to each fog node, which can be reflected by the CPU occupancy across time, as illustrated in Figure 4.7b. In addition, we also observe that by down-tuning the selectiveness from $\rho(x) = x^{10}$ (blue dashed line with triangle markers) to $\rho(x) = x$ (red dashed line with round markers), the average CPU occupancy is reduced. This means processing loads are divided more evenly among fog nodes (red bars in the inner box of Figure 4.7), instead of overloading a small number of fog nodes (blue bars concentrating at the high occupancy end).

Finally, we examine the total device and network efforts of the fog-computing system, considering the offloading process of all tasks generated during the same period of time, in Figure 4.8. As illustrated by the linear dashed and dotted lines, both the total device effort and the network effort are linear with respect to the network size N, which can be obtained from Corollary 4.2. An interesting observation is the impact of selectiveness on the network efforts. When task nodes are more 'picky', or reluctant to offloading adjustments, the fog system can benefit



(a) Total device effort.

(b) Total network effort.

Figure 4.8 Under the delay-based offloading, the total device and network effort of the fog system (per unit time) are $O(N)$ with respect to network size N.

in the sense that less network effort is spent to offload the tasks (as indicated by the higher blue lines in Figure 4.8b), which indicates tasks are offloaded to fog nodes that are closer to the task node. However, this reduction in network effort is achieved at the cost of burdening a small portion of the fog nodes, as indicated by higher CPU occupancy in Figure 4.7b. Among the linearly growing device and network efforts, the former can be easily provided by introducing more fog nodes, while for the latter, it is rather difficult, or even impossible, to sustain such a system, considering the limited network capacity provided at the network edge.

4.5 Summary

In this chapter, we studied the performance evaluation problem for task offloading processes in the fog, through the lifetime individual tasks, device effort, and network effort defined via data movements. We propose a gravity-based offloading model, by which a variety of offloading criteria can be described. The storage and communication resource demand are then quantified by the device effort and network effort metrics, such that resource consumption under different schemes can be measured through a set of unified metrics. Through analysis and simulation, we find that the time and efforts to offload a single task do not scale with the network size, which indicates that, the total resource consumption for all the data offloading processes scale linearly with the network size, under the gravity-based offloading schemes. As our model can describe various offloading criteria, these results are generally applicable in understanding how multiple tasks move in a resource-constrained system, such as the fog paradigm, as well as the impact (in the forms of resource consumption) on the underlying provision networks.

Chapter 5

Spectrum Dynamics: Modeling, Analysis, and Design of Spectrum Activity Surveillance in DSA-enabled Systems

In this chapter, we study spectrum activity surveillance (SAS) to address *what is the observable impact of mobile data*. SAS is essential to wireless systems, especially those open to dynamic spectrum access (DSA), due to spectrum efficiency concerns. Considering monitoring hardware are actively being developed, we take a modeling approach toward this question, such that the monitor model can be fine-tuned to describe spectrum monitors with different characteristics.

We introduce a three-factor space, composed of *spectrum*, *time*, and *geographic region*, in which spectrum activities are only observable in a *closed* subspace, due to their locality in the three domains. We identify and quantify the two objectives of SAS, based on which we formulate the strategy design problem in two steps: 3D-tessellation for sweep (monitoring) *coverage* and graph walk for detecting *spectrum culprits*, that is, wireless devices responsible for unauthorized spectrum occupancy. To efficiently observe the spectrum dynamics, as the result of mobile data, we design efficient SAS strategies with multiple monitors: a low-switching-cost strategy for systems with dedicated monitors, and performance-guaranteed random strategies for systems with crowd-source monitors. We find that randomized strategies by m monitors (even with limited switching capacities) can achieve a sweep coverage over a space of n assignment points in $\Theta(\frac{n}{m}\ln n)$ time, and detect an oblivious or adversarial spectrum culprit in $\Theta(\frac{n}{m})$ time. This $O(\frac{1}{m})$ scaling indicates that, both SAS objectives can be achieved with a linear 'speedup' as more spectrum monitors can be deployed, which provides performance expectations for existing systems. In addition, size n of the assignment space is obtained as a function of the monitoring power of individual monitors, so the performance bounds in turn outline design requirements for spectrum monitors, in order to achieve a certain level of SAS performance.

5.1 Introduction

dynamic spectrum access (DSA) has been envisioned as a key technology for future high-speed heterogeneous wireless systems [50], *e.g.*, 5G cellular networks [115], since it is expected to boost spectrum efficiency by allowing wireless devices to temporally operate beyond their designated spectrum bands, so as to mitigate the gap between the increasing frequency demand and the crowded licensed radio spectrum. It is important on both individual and system levels: it is essential to advanced cognitive radio (CR) technologies, *e.g.*, CR non-orthogonal multiple access (CR-NOMA) [38]; and it is also preliminary to abstraction of wireless resources in a system, *e.g.*, wireless network virtualization (WNV) [72, 69]. Despite its great potentials, the opportunistic and open nature of DSA bears an intrinsic demand for *SAS*, as both a prerequisite and a supplement to such spectrum-agile systems.

5.1.1 Motivation

An SAS process is expected to carry out continuous scans of spectrum activities on the frequencies of interest, for the purpose of usage data collection and spectrum regulation policing/enforcing. On a systematic level, surveillance logs reflect the spectrum usage in wireless communication systems, and can hence be analyzed for system management, as well as data disclosure [48] purposes; on an individual level, real-time spectrum occupancy status can serve as a crude input to reveal and predict the spectrum sensing range [84] for opportunistic spectrum access, which can reduce the sensing time as well as the access delay. To this end, Google [44] and Microsoft [80] have launched their *spectrum database* projects, which provide availability of TV white space over the entire United States, as a preliminary step toward real-time DSA applications and the construction of *radio environment map* (REM) [53]. In this sense, SAS is expected to act as a 'spectrum-meter', passively recording instantaneous occupancy data of different spectrum slices for future analysis.

On the other hand, as an immediate beneficiary of this opportunistic environment toward higher spectrum efficiency, spectrum *culprit*, which refers to overly-aggressive or malicious users, may undermine the 'right-of-way' of legitimate users, and even downgrading performance of the entire system, by occupying unauthorized frequency bands that are promised to other legitimate users. This problem is especially severe in DSA-enbaled systems with distributed spectrum sharing schemes, where a simple Listen-Before-Talk (LBT) mechanism [121] is preferred due to its scalability and comparable throughput performances. In such systems, it is easy for 'smart' spectrum culprits to abuse the DSA-enabled system, owing to the application of machine learning in cognitive radios [116, 87]. Consequently, SAS is expected to act as the 'spectrum-police', proactively detecting spectrum misuse, guarding the rights of legitimate users, and preserving forensics for further actions.

Therefore, SAS is both a premise to leverage spectrum efficiency in compliance to policy enforcement, and a proactive approach to catch the spectrum culprits. Such a system-level function of a DSA-enabled system is completed by *spectrum monitors*, who take advantage of spectrum sensing, networking, and data processing techniques, to collect spectrum occupancy measurements, and identify spectrum culprits based on collected data. In other words, a spectrum monitor is logically composed of three building blocks: sensing hardware, measurement/detection algorithm, and communication protocol. For a large-scale commercial DSA-enabled system, *e.g.*, a multi-operator LTE-WiFi overlay network in the 5 GHz unlicensed frequency bands described in [121], it is necessary to include, and coordinate multiple monitors to provide reliable and timely-updated SAS results.

5.1.2 Related Work

In this regard, existing literature on SAS can be broadly summarized into two categories: single-monitor technique and multiple-monitor orchestration. The former develops prototypes [91, 84], technique and algorithms [71, 132], for individual spectrum monitors, that can effectively differentiate spectrum misuse or abnormalities from normal activities, *e.g.*, statistical significance testing [71], and the spectrum permit mechanism [132], for security enhancement and attack mitigation. In contrast, the latter focuses on efficient deployment and cooperation of *multiple* monitors for the purpose of better surveillance coverage [107], lower switching cost [56], or faster detection of culprits [65]. To this end, spectrum occupancy measurement and interference map construction with commercial dedicated monitors has been studied in [53], while the crowd-source sensing/monitoring paradigm is proposed for cost and flexibility improvement, taking advantage of collaboration [56] and distributed data decoding [37].

In prior studies of multiple monitor deployment strategies (*e.g.*, [107, 56, 65]), an implicit assumption is made for spectrum monitors to be sufficiently powerful, such that they can watch over the entire geographical region of interest and tune/move without any limit. The fact, however, is that most spectrum activities, including communications, attacks/jamming and monitoring/sniffing, are *local*, *i.e.*, confined in both the frequency domain and the space domain during a fixed-length time interval, as noted in prototype design [91], and spectrum occupancy measurements [53]. This discrepancy is especially pronounced in wide-band wide-area monitoring, *e.g.*, spectrum database or REM construction, which naturally leads to an open question: *how to perform an spectrum activity surveillance (SAS)) process and design SAS strategies (with multiple monitors) for DSA-enabled systems*?

5.1.3 Our Approach and Contributions

Hindered by the constraints on spectrum license and high deployment expenses, studying the SAS problem via field tests is not a viable option, especially at the early stage when development of prototypes [91, 84], as well as standardization for CR and DSA, are still underway. Therefore, taking into consideration various monitor settings and SAS scenarios, we take a modeling approach to study SAS processes from perspectives of surveillance coverage and culprit detection. Seemingly trivial, the SAS problem is actually challenging in the following aspects.

First, objectives of SAS, such as data collection and culprits detection, are by-and-large global and collective, lacking a consolidated measure, through which a monitor deployment strategy can be fairly evaluated. Second, if spectrum is considered as a 1-D domain, the surveillance problem over a geographical region is naturally extended to a 3-D space, in which tracking surveillance coverage and analysis are both non-trivial.

To address these challenges, we construct a *spectra-location space* that incorporates spectra, temporal and geographical domains, in which the locality of spectrum activities are captured by limited range and closed spaces. With respect to the modeling, design, and analysis of a SAS process, our contributions can be summarized as follows:

We formally define *monitoring power*, *switching cost*, and *switching capacity* to characterize monitors' and culprits' spectrum activities, and formulate the SAS process into a tractable graph walk process with space-tessellation, such that a collective surveillance function are transformed into localized (even distributed) actions of individual monitors.

We translate the qualitative data collection and culprit detection objectives of SAS processes into two quantitative metrics in the time domain, *i.e.*, the *coverage time* and *detection time*, such that different SAS (monitors deployment) strategies can be evaluated, and fairly compared.

We present a deterministic SAS strategy with low switching cost for systems with dedicated spectrum monitors, and randomized strategies specialized to protect against adversarial spectrum culprits, which is suitable for crowd-source surveillance scenarios. Despite the switching capacity limit, randomized strategies of m monitors can achieve a full sweep coverage over a spectra-location space of n assignment points in $\Theta(\frac{n}{m}\ln n)$ time, and detect a persistent or adversarial culprit in $\Theta(\frac{n}{m})$ time.

5.2 Problem Formulation

In this section, we formally define the spectra-location space, spectrum activities and performance metrics to formulate the SAS problem.

5.2.1 System Model

Let time t proceed in discrete steps, *i.e.*, $t \in \mathcal{T} = \{1, 2, \cdots\}$. Consider a DSA-enabled system deployed in a geographical region $\mathcal{A} \subset \mathbb{R}^2$. The spectrum of interest, $\mathcal{S}$, refers to the spectrum blocks that are shared[1] among K radio access technologies $\{\text{RAT}_i\}_{i=1}^{K}$ allowed in this system.

5.2.1.1 Spectra of Interest $\mathcal{S}$

Each RAT_i has a licensed band LB_i exclusively reserved for authorized RAT_i users, and an unlicensed band UB_i to be shared with users accessing via other RAT's. Each LB_i or UB_i can be

[1] There are two spectrum-sharing scopes for a DSA-enabled system: the inter-technology DSA, which only shares the unlicensed spectrum bands [26], and the *spectrum commons*, in which licensed bands are also included, and each device has equal spectrum access right on a cost basis [121]. Both scopes can be described by our model.

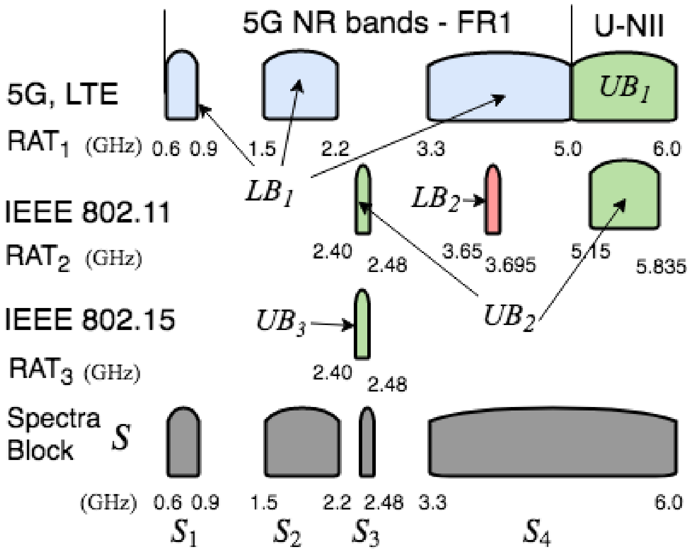


Figure 5.1 An example of the spectra block $\mathcal{S}$ in sub-6GHz frequency bands.

viewed as an *interval* identified by the lowest and highest frequency as its endpoints (or a union of such intervals), then the union of all licensed and unlicensed bands, $\mathcal{S} := \cup_{i=1}^{K}\{\text{LB}_i \ \cup \ \text{UB}_i\}$, is the target of a SAS process in a DSA-enabled system.

An example of spectral block is shown in Figure 5.1, where $K = 3$ RAT's are allowed in the system: cellular (LTE/5g, RAT_1), IEEE 802.11 (WiFi, RAT_2), and IEEE 802.15 (Bluetooth, RAT_3). Among these, RAT_1 has the licensed 5G New Radio (5G NR) FR1 bands [3] to itself, as indicated by LB_1, while its unlicensed U-NII bands UB_1 are shared with RAT_2, on which licensed-assisted LTE access co-exists with WiFi access [121, 26]. Meanwhile, the unlicensed ISM bands UB_3 are shared by RAT_2 and RAT_3. The spectra of interest $\mathcal{S} = \cup_{i=1}^{3} S_i$ is their union.

Without loss of generality[2], we write $\mathcal{S}$ as interval $[s_L, s_H] \subset \mathbb{R}$, and further divide it into $\lceil \frac{s_H - s_L}{\Delta f} \rceil$ *spectrum slices* of width Δf, which is determined by:

(1) *Channel bandwidth* of $\{\text{RAT}_i\}_{i=1}^{K}$. There may not be a unified channel access scheme on $\mathcal{S}$ when $K > 1$. For instance, the U-NII bands can be accessed through LTE and WiFi. Under the former, the standard channel bandwidth are 1.4, 3, 5, 10, 15, and 20 MHz, while under the latter, the channel bandwidth ranges from 10 to 160 MHz. Further, an LTE channel is divided into resource blocks (180 KHz) that contain 12 sub-carriers, while each IEEE 802.11n channel (20 MHz) contains 52 sub-carriers that is of 312.5 KHz wide. Therefore, we choose the slice width Δf as a common divisor of all the channel bandwidths allowed by the K RAT's, such that a channel under each RAT_i contains k_i spectrum slices, where $k_i \in \mathbb{N}^+$ is a positive integer.

[2] Spectra block $\mathcal{S}$ in an wireless overlay system may not form one single continuous interval, rather, it is the union of several non-overlapping continuous intervals, *i.e.*, $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots$. We focus on one of those intervals in this chapter, for the simplicity of notation and understanding.

(2) *Resolution bandwidth* of monitoring devices. Due to the different sampling rates of commercial/prototype monitoring hardware, *e.g.*, 10 MS/s for USRP E310, and 2.4 MS/s for the low-cost SDR prototype designed in [91], resolution bandwidth of spectrum monitors are subject to various limits. Typically, it is set to be 1% to 3% of the channel bandwidth [1, 53] for observable results, but it is also required to be greater than 1 KHz to avoid overloading [1].

Based on these, a spectrum slice of width Δf will be used as the smallest[3] unit of spectrum trunk to be associated with an *access specification* and a time-varying observable *state*.

For each spectrum slice, an *access specifications* states the legitimate way to access this slice, including allowed RAT, maximum transmitting power, maximum aggregated channel bandwidth, register/authentication procedure, and so on. For example, an 1 KHz slice in the 5G NR FR1 (LB_1 in Figure 5.1) can only be accessed through LTE/5G, with transmission settings specified in 3GPP technical specifications, *e.g.*, [3]. In this way, spectra block S is a database with $\lceil \frac{s_H - s_L}{\Delta f} \rceil$ items, against which monitors checks activities on each slice.

The *state* of a spectrum slice $i \in [1, \lceil \frac{s_H - s_L}{\Delta f} \rceil]$ (frequency range $[s_L + (i-1)\Delta f, s_L + i\Delta f)$) is the result of spectrum activities on this particular slice. Slice i is:

(1) *Idle*, when it is not occupied by any user in the current time step, *e.g.*, the white slice $i = 4$ and slice $i = 5$ in time step $t = 1$ in Figure 5.2.

(2) *Rightfully occupied*, if it is accessed by a device obeying the access specification of this slice. For instance, the blue slices $\{17, 18\}$ are rightfully occupied by an authenticated primary user (PU) at time $t = 2$; the green slice 8 and slices $\{12, 13\}$ (with ✔ marker) are accessed by authorized secondary user (SU) at time $t = 1$.

(3) *Illegitimately occupied*, if the occupant does not comply with the access specification of slice i. For instance, the purple slices $\{11, 14\}$ are beyond the designated bands of the aggressive PU at time $t = 3$; the red slices (with restriction sign) are used by an unauthorized SU during $t = 1$ to $t = 3$; the yellow slices $[1, 6]$ are jammed by an attacker emitting a high-power signal at $t = 2$. We refer to these illegitimate occupants as *spectrum culprits*, to be detected by monitors.

Rightful or illegitimate, all aforementioned spectrum activities take place in a 3-D space that spans over both the spectrum domain and the geographical domain.

5.2.1.2 The Spectra-location Space X

Consider the monitoring process of 1-D spectrum $\mathcal{S}$ over a closed 2-D geographical region $\mathcal{A} \subset \mathbb{R}^2$. Together, they compose a 3-D product space $\mathcal{S} \times \mathcal{A}$, referred to as the *spectra-location space* X. Then for any point $x \in X$, there exist projection maps $p_A : X \to \mathcal{A}$ and $p_S : X \to \mathcal{S}$ that identify the frequency and location of any point $x \in X$ respectively. In the product space X, the *spectra-location distance* d_{SA} between point x_i and point $x_j \in X$ is defined as the product metric induced by the Euclidean distance metrics in the spectrum domain $\mathcal{S}$ and the

[3] Note that Δf is not the frequency range that can be scanned by a monitor during a time step. For example, in Figure 5.2 a monitor (red/blue box) can determine the states of 4 spectrum slices.

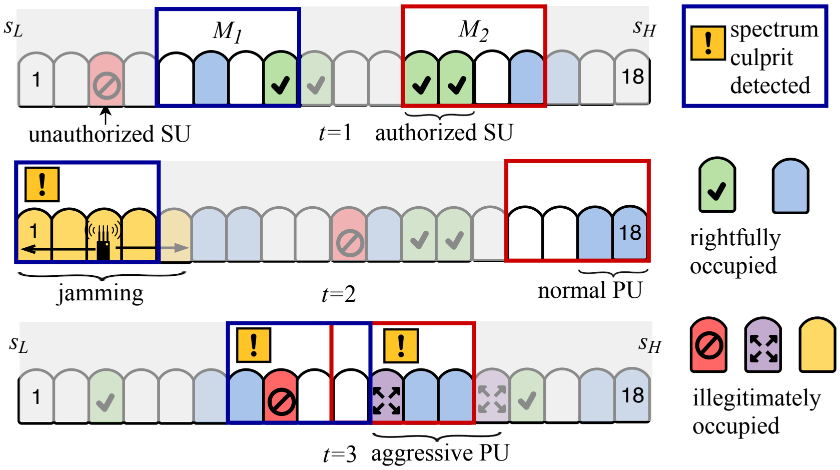


Figure 5.2 Two spectrum monitors M_1 and M_2 watch over spectrum block $\mathcal{S} = [s_L, s_H]$.

space domain $\mathcal{A}$, that is,

$$d_{SA}(x_i, x_j) := ||d_S\left(p_S(x_i), p_S(x_j)\right), \frac{1}{\epsilon} d_A\left(p_A(x_i), p_A(x_j)\right)||_2, \tag{5.1}$$

where $||\cdot||_p$ is the p norm, and $\epsilon > 0$ is a scaling coefficient so that distance d_A and distance d_s are quantitatively comparable [4].

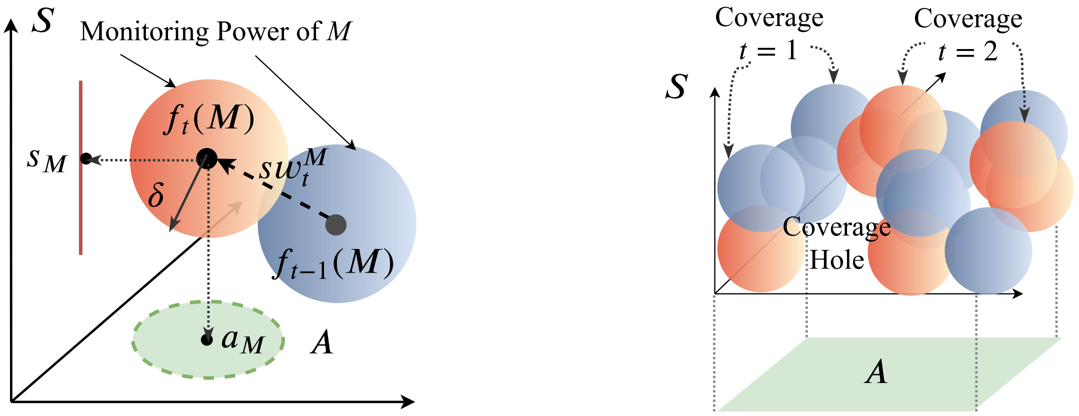
In this sense, the example illustrated in Figure 5.2 is a special case when region $\mathcal{A}$ shrinks to a point $\{a\}$, such that merely the spectrum domain $\mathcal{S}$ needs to be taken into consideration. But still, any monitor can scan, examine and/or record the spectrum activities that take place in a 'box' of four spectrum slices, as a result of which the unauthorized SU is not detected until time $t = 3$. When the space domain $\mathcal{A}$ is sufficiently large, spectrum slices are annotated with locations, due to possible frequency re-use, and a spectrum activity is only observable within the sensing/detection range [53] of a monitor in both spectrum domain $\mathcal{S}$ and geographical space domain $\mathcal{A}$, as described in the following surveillance model.

5.2.1.3 Surveillance Model

Denote $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$ as the set of m monitors in the system. For a single monitor, it can only determine the states of adjacent slices [53] during a time step, as illustrated by the boxes in Figure 5.2. Moreover, due to the attenuation of wireless signal over distance, such constraint also exists in the space domain $\mathcal{A}$. As a result, the capability of a monitor to observe spectrum activities is restricted to a closed space, defined as the *monitoring power*.

Definition 5.1. *For a monitor $M \in \mathcal{M}$ assigned at location $a_M^t \in \mathcal{A}$ and center frequency $s_M^t \in \mathcal{S}$ at time t, the* ***monitoring power*** *of monitor M is defined as a δ-ball centered at*

[4] By comparable, we mean their impacts on monitoring strategy design, *e.g.*, easiness to relocate, energy or time consumption, are on the same order in quantity.



(a) $q(\delta)$-Monitoring power and switching sw_t^M. **(b)** Instantaneous coverage and sweep coverage.

Figure 5.3 Monitoring power, switching, and strategy coverage in spectra-location space $X = \mathcal{S} \times \mathcal{A}$.

$s_M^t \times a_M^t \in X$, *that is,*

$$Ball_\delta(s_M^t \times a_M^t) := \{x \in X | d_{SA}(s_M^t \times a_M^t, x) \leq \delta\}, \tag{5.2}$$

inside which spectrum activities (rightful/illegitimate occupancy) can be identified by monitor M with probability q.

In this definition, Parameter δ and q capture the *locality* and the probabilistic nature of spectrum monitoring activities, respectively. For a spectrum monitor, these two parameters are determined by hardware performances, including sensitivity, noise floor, input range etc. and the detection algorithm. By fine-tuning function $q(\cdot)$, radius δ, and parameter ϵ in Eq. (5.1), a variety of monitoring techniques can be depicted by this δ-ball model.

The *closed* δ-ball describes the surveillance range of a single monitor, as illustrated in Figure 5.3a. This ball shape comes from the hardware constraint on sampling rates [137, 91]. To be more specific, the number of samples that a monitor can collect per unit time is *limited* [53], and these samples can either be employed to cover either a larger bandwidth (large d_S) with a lower sensitivity, or a narrower bandwidth with a higher sensitivity. For instance, in the most commonly-used energy detection method [71, 91], lower sensitivity translates to a higher power threshold, resulting in a reduced detecting range in geographical domain, *i.e.*, small d_A. In other words, for a set of monitoring hardware and a fixed time interval, large d_S and large d_A can not be achieved simultaneously, hence the closed ball shape.

Function $q : \mathbb{R} \to [0, 1]$ quantifies the reliability of monitoring results within the monitoring power, or equivalently the detection probability[5] of spectrum culprits. It has the following properties: i) q is a surjective; ii) q is non-decreasing in $\mathbb{R}$; and iii) we can define its inversion q^{-1} :

[5] For any given radius δ, it is always more probable to determine whether a spectrum slice at a location is occupied or not $(q_c(\delta))$, than determining whether the occupancy is legit $(q_c(\delta))$. Consequently, $q_c(\delta)$ for occupancy measurement is greater than or equal to that for culprit detection $q_d(\delta)$. We set $\delta = \max\{\delta > 0 \mid q_c(\delta) = 1\}$, and $q = q_d(\delta)$, such that occupancy measurement is accurate while culprit detection is probabilistic.

$[0, 1] \to \mathbb{R}$ as $q^{-1}(y) := \sup_{x>0}\{q(x) = y\}$, so for any required reliability $y \in [0, 1]$, there exists a critical radius $\delta^* = q^{-1}(y)$, above which the detection results are not acceptable. Consequently, if a point $x \in X$ is covered by the $q(\delta)$-monitoring power of n monitors, illegitimate occupancy at this point can be detected with a higher probability, $1 - [1 - q(\delta)]^n$.

In the remaining of this chapter, radius δ in our model refers to $\delta_* = q^{-1}(1)$ that can guarantee a fully reliable detection result, if q is not explicitly specified.

5.2.1.4 Exploit Model

Recall a spectrum culprit at time t is the occupant of a spectrum slice that does not comply with the access specifications, as exemplified in Figure 5.2. Consider a spectrum culprit $R \in \mathcal{R}$ located at $a_R \in \mathcal{A}$, who illegitimately occupies one or multiple spectrum slices, denote as $S_R \subset \mathcal{S}$, at time t. Culprit R leaves a 'mark' $R_t = S_R \times a_R \subset X$. The wider S_R is, the larger the 'mark', and the more detectable R becomes. For analysis reasons, we consider spectrum culprits that are most difficult to detect as the worst-case scenario, that is, S_R shrinks to a point $\{s_R\} \in \mathcal{S}$, such that we can write $R_t \in X$; we also assume that, throughout time $\mathcal{T}$, culprits stays in the system and continues its spectrum exploit[6].

Then over time $\mathcal{T}$, the exploit marks of culprit R constitute an *exploit sequence* $\{R_t\}_{t\in\mathcal{T}}$. The detailed exploit *pattern* of spectrum culprits, *i.e.*, how a spectrum culprit $R \in \mathcal{R}$ assigns its exploit sequence, can be either oblivious or adversarial, depending on its learning capability, and will be discussed later in details. At any time t, culprit R is considered to be *detectable* with probability at least $q(\delta)$, if the exploit mark R_t overlaps with the monitoring power of some monitors, *i.e.*, $\exists M_i \in \mathcal{M}$ such that $R_t \cap Ball_\delta(f_t(M_i)) \neq \phi$.

5.2.1.5 Switching Model

Despite their different roles in the DSA-enabled system, normal users (denoted as $\mathcal{U}$), spectrum culprits ($\mathcal{R}$), and monitors ($\mathcal{M}$), are all wireless devices capable of moving and tuning. Both actions will result in a relocation of devices in the spectra-location space X, which we refer to as a *switching* sw_t^Y of device $Y \in \mathcal{U} \cup \mathcal{M} \cup \mathcal{R}$, that is, a move of Y from point $Y_{t-1} \in X$ to point $Y_t \in X$. For example, in Figure 5.3a, the switching sw_t^M corresponds to the relocation and tuning of monitor M between $t-1$ and t, hence the change of its monitoring power (from the blue ball to the red ball). As opposed to the assumption in [91, 65], this common switching action is also constrained in X, due to the induced *cost* in the form of time, energy, budget, and so on, which is a design concern for SAS process in some scenarios. Therefore, we define the *switching capacity* to capture this constraint in space X.

[6] In reality, it is possible that a culprit occasionally misuse some spectrum slices, or even leave the system indefinitely after such exploit. To quantify the performance of a SAS system based on its ability to catch these culprits would not be objective, or fair, as it depends on the culprits' spontaneous behavior. Therefore, we assume culprits stays in the system across $\mathcal{T}$. On the other hand, whenever a culprit's misconduct is recorded for the first time, some attributes, *e.g.*, spectrum fingerprint [12], can be subtracted and distributed to all monitors, such that a match can be found even if the culprit has been accessing the spectrum legitimately ever since.

Definition 5.2. *Let $Y_t \in X$ denote the location (in space X) of device Y at time t, the* ***Switching Capacity*** *α_Y of Y is defined as the maximum distance in X, that device Y can switch over by one action in a time step, that is,*

$$\alpha_Y := \sup_{t \in \mathcal{T}} \left\{ d_{SA} \left(Y_t, Y_{t+1} \right) \right\}. \tag{5.3}$$

Device Y is referred to as an α_Y-monitor or α_Y-culprit.

The switching capacity limit can take different values in different SAS scenarios. For a dedicated (physical) monitor, such as specialty monitoring hardware mounted on a drone or vehicle, a switching action is composed of physical movement and/or tuning. Consequently strategy design is restricted by a quantitative switching cost, including time, energy, budget *etc*, which is a function of the switching distances $d_{SA}(\cdot, \cdot)$. On the contrary, switching actions in a crowd-source scenario are merely changes of surrogate (logical) monitors, which are wireless devices that are capable of and willing to monitor the spectrum activities for the system. Consequently, if immediate communication among participants is guaranteed, or there exists a central controller capable of timely coordination, switching will not be constrained, *i.e.*, $\alpha_M = \infty$; otherwise for the case of distributed control, which relies on local wireless communication, switching may not not possible beyond the communication range of participating devices. Designing SAS strategies for these two scenarios are discussed in Section 5.4 and Section 5.5, respectively.

5.2.2 Performance Metrics and Strategy Design Problem for SAS

To formulate the SAS strategy design problem, we first need to formally design a *strategy* with respect to the m monitors. At the beginning of time step t, each monitor $M_i \in \mathcal{M}$ is assigned to a spectra-location point $f_t^m(M_i) \in X$, through an *assignment* map $f_t^m : \mathcal{M} \to X^m$, *e.g.*, f_1 and f_2 in Figure 5.3a. Allowing time t to proceed in $\mathcal{T}$, assignment points of the m monitors constitute a *strategy*.

Definition 5.3. ***(Strategy***[7] *f_t^m A strategy $\{f_t^m\}_{t \leq T}$ is a sequence of assignments during time interval $[1, T] \subset \mathcal{T}$, subject to the switching capacity constraints $\{\alpha_{M_i}\}_{M_i \in \mathcal{M}}$. During time step t, monitors in set $\mathcal{M}$ can scan $C(f_t^m) = \bigcup_{M_i \in \mathcal{M}} Ball_\delta \left(f_t^m(M_i) \right)$ en masse, which is referred to the (surveillance) coverage of assignment f_t^m. Sweep-coverage of a strategy $\mathcal{C}(f)$ is then the union of sequence $\{C(f_t^m)\}_{t \in [1,T]}$.*

Recall the two objectives of a SAS process, that is, spectrum occupancy measurement and culprit detection. The former urges for a quick sweep-scan of the entire spectra-location space X, *i.e.*, minimizing the time needed to satisfy the coverage goal $X \subset \mathcal{C}(f)$, such that spectrum (occupancy) status can be timely recorded and updated to users. The latter requires effective

[7] Superscript m in f_t^m denotes the number of monitors, while subscript t denotes time. A second subscript may be added to differentiate strategy types, *e.g.*, $f_{S,t}^m$ for deterministic strategy. Any of the three denotations (number of monitors, time and type) may be omitted, when no confusion is raised.

detection of spectrum culprits, such that the time that an undetected culprit illegitimately occupies spectrum slices can be reduced. For instance, in the special case scenario ($X = \mathcal{S} \times \{a\}$) illustrated in Figure 5.2, the entire X is sweep-covered at $t = 3$, and the unauthorized SU (culprit) exploited the system for two time steps before its detection at $t = 3$. In other words, the efficacy of a strategy can be quantitatively evaluated and fairly compared through the following two temporal metrics, with respective to the coverage and detection goals.

Definition 5.4. *Under strategy $\{f_t^m\}_{t\in T}$, the* ***coverage time*** *T_f^m is defined as the first time that its sweep-coverage $\mathcal{C}_T(f^m)$ contains every point in space $X = \mathcal{S} \times \mathcal{A}$, that is,*

$$T_f^m := \min\{T \in \mathcal{T} \mid x \in \mathcal{C}_T(f^m),\ \forall x \in X\}. \tag{5.4}$$

The ***detection time*** *$\tau_R(f^m)$ of a culprit R with exploit sequence $\{R(t)\}_{t\in T}$, is defined as the first time that culprit R can be identified by any of the m monitors, that is,*

$$\tau_R(f^m) := \min\{t \in \mathcal{T} \mid \left[\sum_{i=1}^{m} \mathbb{1}_{R_t \in Ball_\delta(f_t^m(M_i))} D_i\right] \geq 1\}, \tag{5.5}$$

where detection outcome D_i is a Bernoulli r.v. with mean q.

Particularly for $\delta = \delta_*$, when monitoring (detection) result is fully reliable, *i.e.* $q(\delta_*) = 1$, the detection time can be further simplified to

$$\tau_R(f^m) := \min\{t \geq 1 \mid R(t) \in C(f_t^m)\}. \tag{5.6}$$

With the proposed model, this chapter studies the SAS process for a set of$m$ α_M-monitors to achieve the sweep-coverage and culprit detection goals in the spectra-location space X. Specifically, we intend to design strategies $\{f_t^m\}_{t\in\mathcal{T}} \in \{X^m\}^{\mathcal{T}}$ for dedicated and crowd-source SAS scenarios respectively, and examine their *efficacy* by answering the following questions:

- What is the the coverage time T_f of the designed strategy f^m, by which time spectra-location space X is sweep-covered, *i.e.*, $X \subset \mathcal{C}(f^m)$?
- Under strategy f, what is the detection time $\tau_R(f^m)$ of a spectrum culprit $R \in \mathcal{R}$ with the exploit sequence $\{R_t\}_{t\in[1,\mathcal{T}]}$?

5.3 A Two-step Solution

Designing an SAS strategy $\{f_t^m\}_{t\in\mathcal{T}} \in \{X^m\}^{\mathcal{T}}$ is equivalent to finding a sequence of assignment points in the spectra-location space X for every monitor $M \in \mathcal{M}$, at every time instant t, subjecting to the switching capacity constraint α_M. There are two major challenges in this process: First, for every time t, the solution space X^m is of infinite size, which hinders both the analysis-based approach and the search-based experiment approach. Second, switching actions of monitors, *i.e.*, the tuning (a move in the spectrum domain $\mathcal{S}$) and/or relocation (a move in the geographical space domain $\mathcal{A}$), can also be constrained by the accompanied cost.

To overcome these challenges, we propose a two-step solution: first, the continuous strategy space $\{X^m\}^{\mathcal{T}}$ is reduced to a discrete and *finite* space $\{V^m\}^{\mathcal{T}}$ through space-tessellation; then any surveillance strategy is formulated as a *walk* on the graph, whose edges illustrate possible switching actions of monitors. In this way, SAS as a global activity is transformed into a chain of individual actions, *i.e.*, switching (walking) of monitors and culprits, such that design of SAS strategies becomes tractable in both dedicated and crowd-source scenarios.

5.3.1 Space Tessellation: Reducing the Solution Space

Driven by the sweep-coverage and culprit detection objectives, the assignment points of monitors $\{f_t(\mathcal{M})\}_{t\in[1,T_f]}$ of a good SAS strategy should have the following properties:

(1) *Least points.* To timely update the spectrum occupancy data, monitors are expected to sweep-scan the entire spectra-location space X as quickly as possible, which translate to achieving the coverage goal $X \subset \cup_{t=1}^{T_f} C(f_t)$ with as few $(m * T_f)$ assignment points as possible.

(2) *Minimal overlapping.* To quickly detect culprits, every assignment f_t is expected to cover as much non-overlapping space (large $C(f_t)$) as possible, which translates to a minimal overlapping of monitoring powers during every assignment[8].

These requirements can be jointly satisfied if the continuous space X is divided into a minimum number of non-overlapping *cells*, each covered/contained by a δ-ball, *i.e.* the monitoring power of a monitor, and every assignment map f_t takes value from the set V of cell centers, instead of the entire X. In this way, the first step becomes a *tessellation* problem of space X.

5.3.1.1 Solution to the Space-Tessellation Problem

Space tessellation, or honeycomb, in the 3-D space X, refers to the close packing of 3-D *cells* without overlaps or gaps. So the objective of the tessellation problem is to find the best form of cells, which can jointly cover a fixed X with the least cells.

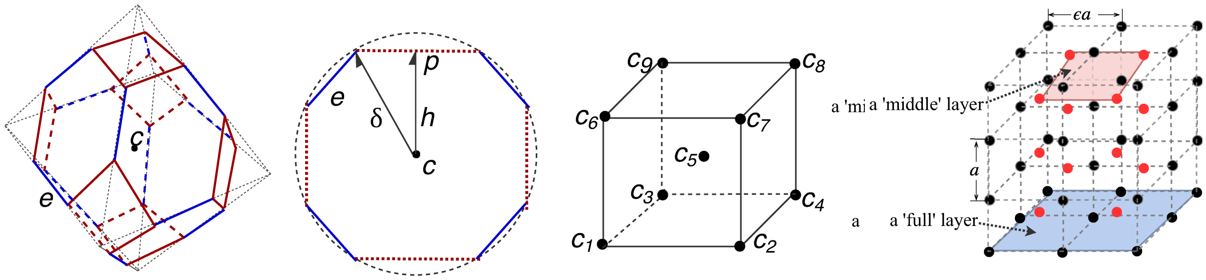
A cell in the tessellation can be regarded as the non-overlapping part of a δ-ball (monitoring power), and comes in various forms in the solution to this classic problem, *e.g.*, cube, hexagon-prism, tetrahedra, etc. The higher volume-efficiency ρ of a cell, that is, the volume ratio of the cell over its insubscribed δ-ball, the more efficient the form of the cell, due to the less overlap between adjacent monitoring power (δ-balls). It would be ideal to fully utilize δ-balls to fill the space. However, direct packing of solid balls always leaves gaps, *i.e.*, spectrum holes in the sweep-coverage, which can be eliminated by pushing the 'elastic' balls into each other. Meanwhile each 'squeezed' balls (the resulting cells) should be inscribed to a δ-ball, *i.e.* the maximum distance of any two points on the cell surface is required to be smaller than 2δ.

[8] Though overlapping monitoring power permits a higher detection probability inside the δ-ball than that of a single one, it is not necessary when the detection probability q is sufficiently high.

[9] Constant $b = \sqrt{\frac{|\mathcal{S}|^2}{4\delta^2 - |\mathcal{S}|^2}}$ for the hexagon-prism cell.

Table 5.1 Volume efficiency of different cell forms discussed in this chapter.

Cell Form	Volume Efficiency ρ	Iso. Quotient[9]	Size $n = \|V\|$
Cube	$\geq \frac{2}{\sqrt{3}\pi} \simeq 0.368$	$\frac{1}{216}$	$\lceil\frac{\sqrt{3}\|\mathcal{S}\|}{2\delta}\rceil \cdot \lceil\frac{3\|\mathcal{A}\|}{8(\epsilon\delta)^2}\rceil$
Hexagon-prism	$\geq \frac{3\sqrt{3}}{2}\alpha\sqrt{(1-\alpha^2)} \leq 0.585$	$\frac{3\sqrt{3}b}{2(3\sqrt{3}+12b)^3}$	$\frac{2\|\mathcal{A}\|}{3\sqrt{3}(1-\alpha^2)(\epsilon\delta)^2}\rceil$
Truncated octahedron	$\geq \frac{24}{5\sqrt{5}\pi} \simeq 0.683$	0.757	$\lceil\frac{5\sqrt{5}\|\mathcal{S}\|\|\mathcal{A}\|}{16\epsilon^2\delta^3}\rceil$



(a) A truncated octahedron centerd at c. **(b)** Top view of the cell. **(c)** Arrangement of cells. **(d)** Cell centers in the Kelvin structure.

Figure 5.4 The discrete assignment space V is composed of cell centers of the Kelvin structure, in the form of 'full' (black) and 'middle' (red) layers.

The space tessellation problem is closely related to the *Kelvin problem* [117], which aims to find the most efficient bubble wrap form to fill a space with the least surface area. In the Kelvin problem, efficiency is quantified by the *isoperimetric quotient*[125], and the ball shape (as the monitor power in our model) has the highest isoperimetric quotient value of 1. Therefore, the most efficient cell form in the tessellation problem is the one that has the highest isoperimetric quotient. In this sense, the best solution for wide (large $|\mathcal{S}|$) spectrum is the *Kelvin structure* (with isoperimetic quotient value 0.757), whose cells are truncated octahedrons (Figure 5.4a-b), arranged in a layered manner (Figure 5.4c-d). The truncated octahedron cell has the highest volume-efficiency ρ compared to other forms, as shown in Table 5.1. Centers of these cells correspond to assignment points composing the discrete and *finite assignment space* V, whose size $n = |V|$ can be obtained from the following proposition 5.1.

Proposition 5.1. *When the spectrum block $\mathcal{S}$ is narrow ($|\mathcal{S}| = 2\alpha\delta$, $0 < \alpha << 1$), the size of the assignment space $n = |V|$ can be determined by tessellation with hexagon-prism cells, i.e.,*

$$n_{hex} = \lceil\frac{2A}{3\sqrt{3}(1-\alpha^2)(\epsilon\delta)^2}\rceil, \tag{5.7}$$

where A denotes the area of region $\mathcal{A}$, δ corresponds to the monitoring power and ϵ is the scaling coefficient in Eq. (5.1)*. Otherwise, size n is achieved by tessellation with truncated octahedron (Kelvin structure) cells, and*

$$n_o \geq \lceil\frac{5\sqrt{5}|\mathcal{S}|A}{16\epsilon^2\delta^3}\rceil. \tag{5.8}$$

Further, if the geographical region $\mathcal{A}$ is rectangular,

$$n_o \geq \lceil \frac{\sqrt{5}|\mathcal{S}|}{4\delta} \rceil \cdot \lceil \frac{5A + 2\sqrt{5A}\delta(3-2\epsilon) + 4\delta^2(1-\epsilon)^2}{8(\epsilon\delta)^2} \rceil, \tag{5.9}$$

where $e = \frac{2}{\sqrt{10}}\delta$ is the edge length of cells.

Proof. As shown in Figure 5.5, the narrow spectrum block, *i.e.*, $|\mathcal{S}| = 2\alpha\delta$, implies that one layer of δ-balls is enough to cover the entire spectrum $\mathcal{S}$, if all of the monitor radios are tuned to the center frequency $s = \frac{s_{max}+s_{min}}{2}$. At s_L and s_H, each δ-ball creates a disk print with radius r on the surface of X, as shown in blue shade in Figure 5.5. Then the 3-D tessellation problem becomes the coverage of a planar area $\mathcal{A}$ with disks of radius $r = \sqrt{(1-\alpha^2)}\delta$, as illustrated in the right of Figure 5.5. The effective non-overlapping part (a cell) is a hexagonal prism.

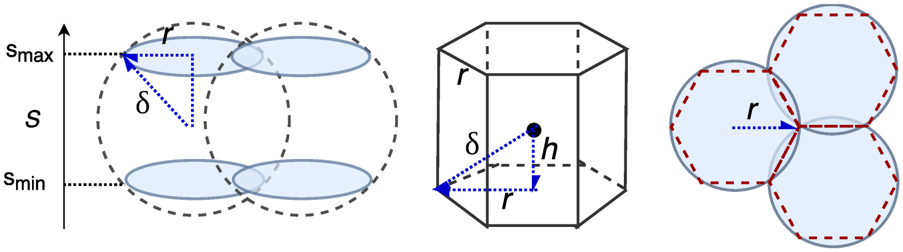


Figure 5.5 Tessellation of X with hexagonal prism when spectrum range is narrow.

To cover area $\mathcal{A}$ with the least overlap among disks, the disks should be arranged according to the hexagonal tessellation, with each hexagon inscribed to a circle, as in Figure 5.5 right. The edge length of any hexagon $r = \sqrt{(1-\alpha^2)}\delta$. Bound on the number of assignment points n_{hex} in Eq. (5.7) follows from hexagonal geometry. Also, the coverage of the boundary points (the top and bottom surface) are guaranteed.

For broader spectrum $\mathcal{S}$, the best known 3-d tessellation form is the Weaire-Phelan structure[125], which has two types of cells with the same isoperimetic quotient value 0.764. However, the arrangement of two types of cells make it more difficult to design monitor assignment strategies. Therefore, we adopt the sub-optimal *Kelvin structure*, whose cell is a truncated octahedron with isoperimetic quotient value 0.757[10], as shown in Figure 5.4a. Each cell is a polyhydron with six square faces and eight hexagonal faces, both with edge length e. The center of each cell coincide with the center c of its outscribed δ-ball, so its largest cross section through the center c is an octagon with edge length $p\sqrt{2}e$.

Since $e = \frac{2}{\sqrt{10}}\delta$, the volume efficiency $\rho_{oct} = \frac{24}{5\sqrt{5}\pi} \simeq 0.683$, much higher than the hexagon prism cell and the cubic cell, as listed in Table 5.1. To form a 3-d tessellation, the arrangement pattern of cells is body-centered cubic, as shown in Figure 5.4c-d. Then lower bound of n in

[10] The Kelvin structure is allowed to have curved surfaces to have a smaller surface area, which is not necessary in our case.

Eq.(5.8) can be obtained by

$$n_o \geq \frac{|\mathcal{S}||\mathcal{A}|}{\epsilon^2 \mathrm{Vo}^o} = \lceil \frac{5\sqrt{5}|\mathcal{S}|A}{16\epsilon^2\delta^3} \rceil.$$

A tighter bound can be found if $\mathcal{A}$ is a rectangular region. Suppose the length of $\mathcal{A}$ is l, and the width is w. The number of "full" layers (blue shade in Fig 5.4 right) is

$$L_{full} = \lceil \frac{|S|}{2h} + 1 \rceil = \lceil \frac{\sqrt{5}|S|}{4\delta} \rceil + 1, \tag{5.10}$$

and the number of "middle" layers (red shade in Fig 5.4 right) is $L_{mid} = L_{full} - 1$. For the coverage of region $\mathcal{A}$, the number of cells in a "full" layer can be lower bounded by

$$N_{full} = \lceil \frac{l + \sqrt{2}e}{2h \cdot \epsilon} \rceil \cdot \lceil \frac{w + \sqrt{2}e}{2h \cdot \epsilon} \rceil \geq \lceil \frac{5A + 4\sqrt{5A}\delta + 4\delta^2}{16(\epsilon\delta)^2} \rceil. \tag{5.11}$$

And the number of cells in a middle layers can be lower bounded by

$$N_{mid} = \lceil \frac{l + \sqrt{2}e}{2h \cdot \epsilon} - 1 \rceil \cdot \lceil \frac{w + \sqrt{2}e}{2h \cdot \epsilon} - 1 \rceil \geq \lceil \frac{5A + 8(1-\epsilon)\sqrt{5A}\delta + 4\delta^2(1-4\epsilon)}{16(\epsilon\delta)^2} \rceil + 1. \tag{5.12}$$

Then the total number of truncated octahedron cells needed is

$$n_o = L_{full}N_{full} + L_{mid}N_{mid}, \tag{5.13}$$

which yields the tighter bound Eq.(5.9) with some manipulations. □

By space-tessellation, sweep-coverage of X is guaranteed as long as the assignment maps $\{f_t\}_t$ are jointly surjective on the *finite* assignment space V of cell centers. In other words, it is sufficient for monitors ($\mathcal{M}$) to switch among assignment points in V, which greatly reduces the size of the solution space (from ∞ to $|V|^m$) for every time step. For the ease of notation and discussion, we can also restrict the range of exploit points to V, because any point x in X can be uniquely mapped to a cell with its center v_x in V, and the probability of a spectrum culprit $R \in \mathcal{R}$ being detected at $R_t \in X$ is the same as that at $R_t = v_x$.

5.3.1.2 Exploit Patterns of Culprits in Assignment Space V

After the preparation of setting up the assignment space V, we categorize the exploit patterns of spectrum culprits in this context, *i.e.*, how a spectrum culprit $R \in \mathcal{R}$ determines its exploit point $R_t \in V$ for the next time step, with respect to its learning capabilities.

Definition 5.5. *A* ***persistent culprit*** $R_p \in \mathcal{R}$ *refers to a spectrum culprit, whose exploit sequence* $\{R_t\}_t$ *does not change over time, that is,* $\{R_t\}_t$ *is composed of i.i.d. r.v.'s* R_t^p*, all distributed with PMF* $g_{R_p}(v)$*, where* v *is an assignment point in* V*.*

Persistent culprit R_p can describe a variety of exploiting strategies with different PMF $g_{R_p}(v)$. For instance, as shown in Table 5.2, R_s is a stationary culprit that can only access a selective range of frequencies; R_{sd} corresponds to a DSA-enabled stationary culprit; and R_{md} is a mobile DSA-enabled culprit that can move in region $\mathcal{A}$.

Table 5.2 Three types of persistent spectrum culprits.

Type	Culprit Description	PMF $g_{R_*}(\cdot)$	Detection Time[11]
R_s	stationary, fix-frequency	$g_{R_s}(v) = \mathbf{1}_{v_s}(v),\ \forall v \in V$	$\mathbb{E}\left(\tau_s(f_S)\right) = \frac{T_s}{2}$
R_{sd}	stationary, DSA-enabled	$g_{R_s}(v) = \begin{cases} \frac{1}{\lvert V_{sd}\rvert}, & \text{if } v \in V_{sd}, \\ 0, & \text{otherwise.} \end{cases}$	$\mathbb{E}\left(\tau_{sd}(f_S)\right) \leq \frac{n}{m} \simeq T_s$
R_{md}	mobile, DSA-enabled	$g_{R_{md}}(v) = \frac{1}{n},\ \forall v \in V$	$\mathbb{E}\left(\tau_{md}(f_S)\right) = \frac{n}{m} \simeq T_s$

As opposed to oblivious persistent culprits, machine learning-assisted RAT [116, 87] allows sophisticated culprits to steer the game toward their benefit, by actively dodging monitors [65]. The intuition behind adversarial culprits is that, once a SAS strategy is known[12] by a culprit R_a, R_a can then switch to points that are less probable to be monitored in the next time step.

Definition 5.6. *An* ***adversarial culprit*** $R_a \in \mathcal{R}$ *is a spectrum culprit with prior knowledge of current strategy* $\{f_t^m\}_{t\in\mathcal{T}}$*, that is,* R_a *knows the set of probabilities* $\{v \in \cup_{M_i} f_t^m(M_i)\}_{v\in V}$ *ahead of time* t*, and determines its current exploit point* R_t^a *with PMF* $g_{R_a}^t(v) = \frac{1}{\lvert \text{Void}(t)\rvert}$ *for point* $v \in \text{Void}(t)$*, where* $\text{Void}(t) = \arg\min_{v\in V} \mathbb{P}\Big(v \in \bigcup_{M_i\in\mathcal{M}} f_t(M_i)\Big)$ *is the 'spectrum hole' that is least likely to be monitored at* t*.*

Note there is no switching constraint in Definition 5.5 and Definition 5.6, which describe the most powerful culprits in terms of switching, *i.e.*, $\alpha_R = \infty$. As discussed before for monitors, it is possible that culprit R has $\alpha_R < \infty$, such that the range of R_{t+1} will be restricted to a smaller subset $N(R_t) = \{v \in V \mid d_{SA}(v, R_t) \leq \alpha_R\}$ of V, with the selecting probability of a point $v \in N(R_t)$ recalculated as $\mathbb{P}(N_{t+1} = v) = \frac{g_R(v)}{\sum_{u\in N(R_t)} g_R(u)}$. However, it is still unclear how range and rate of switching affects the performance of a SAS strategy, which will be addressed in the next subsection, to better formulate the strategy design problem.

5.3.2 Graph Walk: A Chain of Switching Actions

Over the discrete time span $\mathcal{T}$, any SAS process is now a chain of switching actions in the assignment space V. Recall that a switching sw_t^Y is a relocation (tuning in $\mathcal{S}$ and/or movement

[11] The expected detection time of the persistent culprits, including quantity T_s, are discussed in Section 5.4.2.1.

[12] We consider the most powerful culprit (with full knowledge of a strategy) as an extreme case to examine the performance of an SAS system against compromised strategies. In other cases, a weaker culprit can at least observe the long-term visiting probability of any point $v \in V$ as prior knowledge. On the other hand, SAS strategies are required to be disclosed for transparency, such as a crowd-source SAS scenario, which increases the risk of a strategy being leaked to culprits.

in $\mathbf{A}$) of a device $Y \in \mathcal{M} \cup \mathcal{R}$ (monitor or culprit[13]) from time $t-1$ to t, whose range is upper-bounded by the switching capacity α_Y. Next, we discuss switching actions from range (how far) and time (how quick) aspects, to formulate the surveillance process into a graph walk.

5.3.2.1 Range Aspect (Switching Capacity)

The switching range refers to the maximum spectra-location distance d_{SA} over which one switching action is possible. As discussed before, switching actions of dedicated monitors are completed via physical movements and/or tuning of individual monitors, so the switching cost scales with spectra-location distance d_{SA}. For this case, we need a quantitative metric to accurately measure the switching cost for strategy design, which is addressed in Section 5.4, where a low switching-cost monitoring strategy is proposed for dedicated monitors.

On the other hand, switching in a crowd-source monitoring scenario is a change of surrogate monitors, that is, wireless devices (spontaneously) participating in the SAS process. In this case, the switching cost may not scale with the spectra-location distance. Instead, a switching between any two assignment points v_x and $v_y \in V$, can either be possible with a fixed amount of cost (*e.g.*, time and coordination budget), or impossible during one time step. To be more specific, when immediate communication is guaranteed among all participants, a 'handover' between any two monitoring surrogates will be possible in one time step, *i.e.*, $\alpha_M = \infty$; otherwise for distributed crowd-source monitoring that relies on short-range wireless communication to coordinate, any switching action is only possible between two monitors within their communication ranges, *i.e.*, $\alpha_M < \infty$. For crowd-source monitoring scenarios under unlimited ($\alpha_M = \infty$) and limited ($\alpha_M < \infty$), strategy design is discussed in Section 5.5 and Section 5.6, respectively.

5.3.2.2 Time Aspect (Switching Rates)

In addition to switching range, the rate of switching, that is, how many switching actions can be conducted in one time step, is also limited by the hardware constraint. Intuitively, a faster-switching culprit will be more difficult to catch, due to the shorter time that the culprit remains in the monitoring power of any monitors. Counter-intuitively, as we will show in Lemma 5.1, such culprits will be detected in an even shorter period of time.

In Lemma 5.1, consider spectrum monitors with $q(\delta)$-monitoring power, that is, when a culprit shows up in the cell assigned to a monitor (referred to as *co-location*), the probability that it is detected by that monitor during one time step is q. Let $q \cdot p(s)$ ($s \in [0,1]$) denote the detecting probability when the co-location time s is less than one full time step, where the non-decreasing function $0 \leq p(s) \leq 1$ captures the attenuated detection probability due to the decreased co-location time, and has the property of $p(0) = 0$, $p(1) = q$.

[13] Normal users in $\mathcal{U}$ can also switch, but their switching actions do not have any impact on the performance of SAS strategies, only the outcome, and are hence not listed here.

Lemma 5.1. *Suppose culprit R_1 differs from R_2 in only the switching rate: R_1 can switch $k \in \mathbb{N}^+$ times during one time step, while R_2 and monitors can switch once. The detection time of R_1 is stochastically dominated by that of R_2, that is, for any strategy f,*

$$\tau_{R_1}(f) \overset{d}{\leq} \tau_{R_2}(f), \tag{5.14}$$

when the following criterion is satisfied:

$$p(\frac{1}{k}) \geq \frac{1 - [1 - q \cdot \mathbb{P}\big(R^2(t) \in C(f_t^m)\big)]^{\frac{1}{k}}}{q \cdot \mathbb{P}\big(R^2(t) \in C(f_t^m)\big)}. \tag{5.15}$$

Proof. Without loss of generality, suppose the strategy f is carried out by one monitor M. Eq. (5.14) holds trivially for a deterministic strategy f_S, when both R_1 and R_2 are adversarial, because every next to visit assignment point is known by an adversarial culprit. In fact, the detection time $\tau_{R_1}(f_S) = \tau_{R_2}(f_S) = \infty$, which is referred to as the 'wandering hole' problem analyzed in at the end of Section 5.4.

Next we consider the case of f being a randomized strategy, in which the next assignment point is chosen randomly, or R_1 and R_2 being persistent culprits. During a time step t, R_1 generates an exploit sequence $\{R_1^1(t), R_2^1(t), \cdots, R_k^1(t)\}$, and R_2 switches to $R^2(t)$, while monitor M stays at a fixed assignment point $f_t(M) \in V$. The probability that R_2 is detected during t is $Q_2 = q(1 - \mathbb{P}\big(R^2(t) \neq f_t(m)\big))$, while for R_1, the probability of being identified by monitor M is

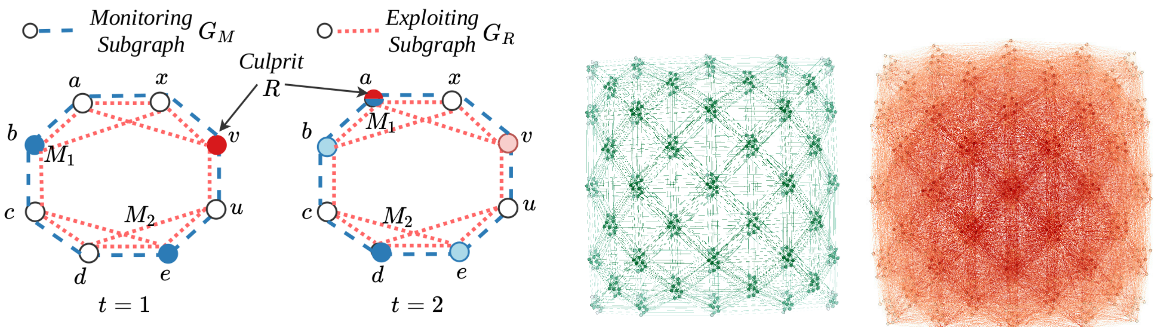
$$Q_1 = 1 - \Pi_{i=1}^k \left(1 - qp(\frac{1}{k})\mathbb{P}\big(R_i^1(t) = f_t(m)\big)\right). \tag{5.16}$$

Given that R_1 and R_2 only differ in switching rates, that is, $\mathbb{P}\big(R_i^1(t) = v\big) = \mathbb{P}\big(R^2(t) = v\big)$, for any $v \in V$ and $i \in \{1, 2, \cdots, k\}$, the probability $Q_1 = 1 - (1 - p(\frac{1}{k})Q_2)^k$. When the condition in Eq. (5.15) holds, we have $Q_1 \geq Q_2$, which means that R_1 is more probable to be detected during any given time step t. Equivalently, the complementary CDF (CCDF) of detection time $R_1(f)$ and $R_2(f)$ satisfy $\mathbb{P}(\tau_{R_1(f)} > l) \leq \mathbb{P}(\tau_{R_2(f)} > l)$, for any integer $l \geq 1$. □

The condition in Eq. (5.15) easily holds when probability $\mathbb{P}\big(R^2(t) \in C(f_t^m)\big)$ is small, *i.e.*, when the size of the assignment space $|V|$ is large enough, such that it is difficult for a culprit to co-locate with any monitor in a single time step. Otherwise, when the assignment space V contains few assignment points, though catching the faster culprit R_1 takes more time than the slower one, the expected detection time can be derived as $\mathbb{E}(\tau_{R_1}) = \frac{1}{kq \cdot p(\frac{1}{k})\mathbb{P}(R_2 \in C(f_t^m))}$, which will not be large, if the attenuation function satisfies $p(\frac{1}{k}) \geq \frac{1}{k}$. Consequently, the difference between the detection time of R_1 and R_2 will be small. Based on this observation, it is reasonable to assume both culprits and monitors switch once in every time step for the rest of this chapter.

5.3.2.3 A Graph Walk on Composite Graph (G_M, G_R)

Accounting switching capacity, the assignment space V is more than a set of points, rather, a subspace that inherits d_{SA} metric from space X. Together with the space-tessellation procedure,



(a) An example of culprit detection process by monitors M_1 and M_2 (blue dots).

(b) Monitoring (green) and exploiting (orange) subgraphs for the simulation scenario in Figure 5.14b.

Figure 5.6 Examples of composite graph (G_M, G_R), illustrating the case of weaker (smaller α_M) monitors v.s. the more powerful (larger α_R) culprit.

the subspace gives rise to a structure that incorporates the possibility of switching actions, *i.e.*, a composite graph, which consists of:

(1) *Monitoring subgraph* $G_M = (V, E_M)$, in which an edge $(u, v) \in E_M$ exists, if and only if $d_{SA}(u, v) \leq \alpha_M$. Then, an arbitrary strategy $\{f_t^m\}_{t \in \mathcal{T}}$ can be seen as a joint *walk* by $m = |\mathcal{M}|$ monitors on the monitoring subgraph G_M.

(2) *Exploiting subgraph* $G_R = (V, E_R)$, in which an edge $(u, v) \in E_M$ exists, if and only if $d_{SA}(u, v) \leq \alpha_R$ of the culprit. Then, the exploiting sequence $\{R_t\}_t$ of R, which contains the assignment points exploited by culprit R, also corresponds to visited vertices of a walk on the exploiting subgraph $G_R = (V, E_R)$.

Graph G_R and G_M have the same vertex set V, and are both sub-graphs of K_n, *i.e.*, the complete graph with $n = |V|$ vertices, which corresponds to SAS scenario of unlimited switching capacities ($\alpha_M = \alpha_R = \infty$). The SAS process, particularly culprit detection, then becomes a graph walk on the composite graph $G = (G_M, G_R)$, in which culprit R is first detected when R and any of the monitors in $\mathcal{M}$, co-locate at an assignment point, *i.e.*, meet on a vertex in G.

Figure 5.6a illustrates an example of the composite graph walk. Monitors $\mathcal{M} = \{M_1, M_2\}$ (solid blue dot) and a culprit R (solid red dot) both reside/walk in the assignment space $V = \{a, b, c, d, e, u, v, x\}$ for two time steps, where the edge sets E_M and E_R are shown in blue dashed and red dotted lines respectively. By $t = 2$, the sweep coverage $\mathcal{C}_t = \{a, b, c, e\} \subset V$, indicating the coverage time $T_f > 2$. The detection time $\tau_R(f) = 1$, because monitor M_1 meets with the culprit R_t at assignment point $a \in V$, during time step $t = 2$.

Formulating the SAS process into a graph walk makes strategy design more tractable, in the sense that the strategy space is now discrete and finite, such that both the theoretic and simulation approaches are viable. Under this formulation, we discuss strategy design and performance evaluation for the dedicated and crowd-source monitoring scenarios, in the following Section 5.4 and Section 5.5, respectively.

5.4 Deterministic SAS Strategies for Dedicated Monitors

Dedicated monitors refer to the specialized monitoring equipment mounted on towers, drones, vans, etc. which are widely used by governmental and commercial agents, *e.g.*, FCC, NTIA, and AT&T, to collect spectrum measurement data [48]. For DSA-enabled systems relying on dedicated monitors, *deterministic* monitoring strategies, that is, monitors traverse a predetermined route to sweep-scan the spectra-location space, are the sensible choice due to its simpleness, *e.g.*, [53]. For such strategies, the *switching cost*, in the form of time, energy or budget, is the key concern in deploying dedicated monitors. The reason behind this is that, switching cost, induced by tuning (in spectrum domain $\mathbf{S}$) and movement (in space domain $\mathbf{A}$), scales with distances in both domains, so it is essential to optimize the strategy for a reduced cost, given that all the monitors are under the control of the SAS function. Therefore, we first define a comprehensive switching cost metric, based on the optimization of which, we propose the low-cost deterministic SAS strategy f_S.

Definition 5.7. *The* ***switching cost*** *from point x_i to $x_j \in X$, is defined as the sum of tuning cost and relocation cost, that is,*

$$\gamma(x_i, x_j) := \beta_S d_S(p_S(x_i), p_S(x_j)) + \beta_A d_A(p_A(x_i), p_A(x_j)), \tag{5.17}$$

where β_S and β_A are cost coefficients for tuning and relocation respectively. The ***cost of strategy*** *$\{f_t^m\}_{t=1}^{T_f}$ is then*

$$\Gamma_f^m := \sum_{t=2}^{T_f^m} \sum_{i=1}^{m} \gamma(f_t^m(M_i), f_{t-1}^m(M_i)), \tag{5.18}$$

where T_f^m is the coverage time defined in Eq. (5.4).

Switching cost Γ_f can be applied to describe time, energy, and budget. For example, the cost coefficients can be set to $\epsilon\beta_A >> \beta_S$ to describe the switching time, since the time it takes a radio head to tune to a different center frequency is approximately 1 ms [54], during which the physical movement of any mobile device is negligible. It could also be the case that re-configuring the center frequency of a radio head is more expensive (in terms of budget) than physical movements, *e.g.*, specialized devices with narrow frequency ranges, such that $\epsilon\beta_A < \beta_S$. Accounting for switching cost between assignment points in space V, the resulting monitoring sub-graph G_M becomes a weighted complete graph K_n, in which the weight of each edge reflects the switching cost along that edge.

5.4.1 Low Cost Deterministic Strategies f_S

Addressing the design concern of switching cost in the dedicated SAS scenario, an *optimal* deterministic strategy is a strategy $\{f_{S,t}\}_{t=1}^{\mathcal{T}}$, whose total switching cost Γ_f is minimized in the solution space $\{V^m\}^{\mathcal{T}}$. In this sense, finding an optimal strategy $\{f_{S,t}^m\}_{t=1}^{T_s}$ is equivalent to finding m vertex-disjoint 'shortest' paths (in terms of switching cost) that jointly cover the assignment space V by time $T_S^m = \lceil \frac{n}{m} \rceil$. This problem is actually an open-path multi-depot multi-travailing salesmen problem (MD-MTSP), which is known to be NP-hard [14, 111].

Observing the structure of assignment space V (as shown in Figure 5.4d) over a rectangular region $\mathbf{A}$, we prove there is an upper bound of the minimum switching cost Γ_{min}^m for small m values, which can be achieved by traversing the space V 'smartly'. Before presenting the main result, we summarize necessary denotations in the following Table 5.3 for the ease of references.

Table 5.3 Parameters and denotations in Theorem 5.1.

Denotation	Meaning	Example in Figure 5.4
L	# of assignment points along the length of $\mathcal{A}$	$L = 3$ (in 5.4d)
D	# of assignment points along the width of $\mathcal{A}$	$D = 3$ (in 5.4d)
H	# of assignment points along the spectra axis $\mathcal{S}$	$H = 4$ (in 5.4d)
$n = \lvert V \rvert$	total # of assignment points in V	$n = 48$ (in 5.4d)
Type-1 edge	edges w/ cost $\gamma_1 := \beta_S a$	$(c_1, c_6), (c_2, c_7)$ (in 5.4c)
Type-2 edge	edges w/ cost $\gamma_2 := \beta_A \epsilon a$	$(c_1, c_2), (c_7, c_8)$ (in 5.4c)
Type-3 edge	edges w/ cost $\gamma_3 := \frac{a}{2}(\sqrt{2}\beta_A \epsilon + \beta_S)$	$(c_1, c_5), (c_5, c_8)$ (in 5.4c)

Theorem 5.1. *For a set of $m \geq 3$ monitors on the assignment space V with parameters L, D and H (defined in Table 5.3), the switching cost Γ_{min}^m can be upper bounded by*

$$\Gamma_{min}^m \leq \begin{cases} \Gamma_*^1 - A_*(m-1), & \text{if } m = 2k+1, \\ \Gamma_*^2 - A_*(m-2), & \text{if } m = 2k+2, \end{cases} \tag{5.19}$$

where $k \in \mathbb{N}^+$, $* = f$ if $\beta_S \geq \frac{2-\sqrt{2}}{2}\beta_A \epsilon$, and $* = g$ otherwise. Quantities Γ_*^1, Γ_*^2 and $A_*(\cdot)$ are determined as

$$\begin{aligned} \Gamma_*^1 &:= K_1\gamma_1 + K_2^*\gamma_2 + K_3^*\gamma_3, \\ \Gamma_f^2 &:= \Gamma_f^1 - (2D-1)\gamma_2 + \gamma_3 + (\frac{H}{2}\gamma_1 + \gamma_2)e(L), \\ \Gamma_g^2 &:= \Gamma_g^1 - 2\gamma_2 - \gamma_1 + 2\gamma_3, \\ A_*(x) &:= \min\{K_1, x\}\gamma_1 + \min\{K_3^*, [x-K_1]^+\}\gamma_3 + [x - K_1 - K_3^*]^+\gamma_2, \end{aligned}$$

where $K_1 = LD(H-1) + (L-1)(D-1)(H-2)$, $K_2^f = (L-1)D + (L-2)(D-1)$, $K_3^f = 2(D-1)$, $K_2^g = L + D - 2$, $K_3^g = 2(L-1)(D-1)$, and $e(L) = 1$ when L is even, $e(L) = 0$ otherwise.

Proof. **(Outline.)** We prove this theorem in a constructive manner, and work our way from the base case when ($m = 1$ or 2) toward the general cases ($m \geq 3$). For the single- or double-monitor cases ($m = 1$ or 2), considering that Type-1, 2 and 3 edges are the least expensive edges in the complete graph K_n in terms of switching cost, the main idea is to construct a traversing route with the most number of least expensive edges (Type-1 or Type-2 depending on quantity

$\beta_S - \frac{2-\sqrt{2}}{2}\beta_A\epsilon$), and as few necessary longer edges as possible. The optimal strategy for these two cases, and their accompanied minimum switching costs Γ^1_{min} and Γ^2_{min}, are presented in Lemma 5.2 and Lemma 5.4, respectively. Then for $m > 2$, the switching cost can be upper bounded by evenly dividing the low-cost routes of $m = 1$ or $m = 2$, and then removing the induced dividing edges, as presented in Lemma 5.3. First, we prove Lemma 5.2 for $m = 1$.

Lemma 5.2. *When $m = 1$, and the assignment space V has L, D and H assignment points along the length, width and height (spectrum $\mathcal{S}$) of X, the switching cost of the optimal strategy Γ^1_{min} satisfies*

$$\Gamma^1_{min} \leq \Gamma^1_* = K_1\gamma_1 + K_2\gamma_2 + K_3\gamma_3, \tag{5.20}$$

where $K_1 = LD(H-1) + (L-1)(D-1)(H-2)$; K_2 and K_3 are determined by the following:

1. *When cost coefficients satisfy $\beta_S \geq \frac{2-\sqrt{2}}{2}\beta_A\epsilon$, $K_2 = (L-1)D + (L-2)(D-1)$ and $K_3 = 2(D-1)$. Then Γ^1_* is achieved by the strategy $f^*_{m=1}$, as shown in Figure 5.7a.*
2. *Otherwise $K_2 = L + D - 2$ and $K_3 = 2(L-1)(D-1)$, achieved by strategy $g^*_{m=1}$, as illustrated in Figure 5.7b.*

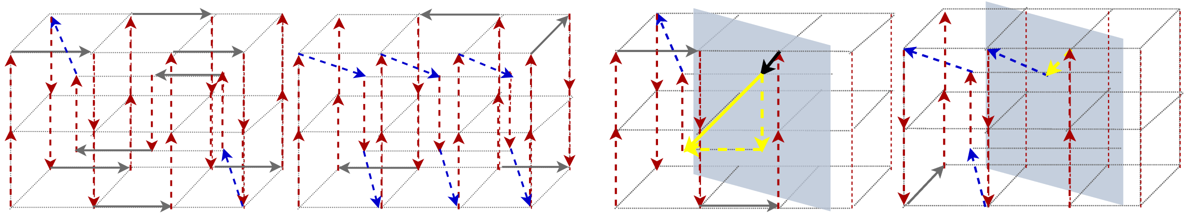

(a) $m = 1$: $f^*_{m=1}$ **(b)** $m = 1$: $g^*_{m=1}$ **(c)** $m = 2$: $f^*_{m=2}(M_1)$ **(d)** $m = 2$: $g^*_{m=2}(M_1)$

Figure 5.7 Low-cost strategies for setting $L = 4,\ D = 2,\ H = 3$. Red, grey and blue indicate Type-1, 2, and 3 edges (defined in Table 5.3), respectively.

Proof. The shortest (in terms of cost) combination of paths connecting the $n = |V|$ points is including all the vertical (Type-1) edges, and then adding the minimum number of necessary type-3 and type-2 edges, as in Figure 5.7a and 5.7b, because the triangle equality holds under the switching cost metric defined in Section 5.4. When $\beta_S \geq \frac{2-\sqrt{2}}{2}\beta_A\epsilon$, the switching cost of the optimal strategy $\Gamma^1_{min} = \Gamma^1_{f^*}$ since the short edges (Type-1 and Type-2) are utilized to the most, indicating $f^*_{m=1}$ (Figure 5.7a) is the optimal strategy because in this case $\gamma_2 \leq \gamma_3$. Otherwise, if $\gamma_2 > \gamma_3$, it is cheaper to use Type-3 edges than Type-2 edges more often, under the circumstance that the total number of edges remains the same, *i.e.* $K_1 + K_2 + K_3 = n - 1$. Therefore strategy $g^*_{m=1}$ (Figure 5.7a) achieves a lower switching cost than strategy $f^*_{m=1}$. □

Now we consider the case of multiple monitors, *i.e.*, $m > 1$.

Lemma 5.3. *When $m > 1$, the minimum cost satisfies*

$$\Gamma^m_{min} \leq \Gamma^1_{f^*} - \min\{K_1, (m-1)\}\gamma_1 - \min\{K_3, [m-1-K_1]^+\}\gamma_3 - [m-1-K_1-K_3]^+\gamma_2,$$

where $[x]^+ = \max\{x, 0\}$, and K_1, K_3, Γ^1_{min} as in Lemma 5.2.

Proof. One possible solution of employing $m > 1$ monitors is to divide the near-optimal path $f^*_{m=1}$ in Lemma 5.2 evenly (in terms of hop count), into m sub-paths, by removing $m-1$ edges. Note that different m will result in the removal of different numbers of Type-1, 2 and 3 edges. However, the minimum cost Γ^m_{min} is always upper bounded by the case that the removal starts from Type-1 edges, which are associated with the minimum switching cost. □

Further, when m is even, a tighter bound can be achieved through an effective way to alter and divide the optimal path in strategy $f^*_{m=1}$ and $g^*_{m=1}$, as shown in Lemma 5.4.

Lemma 5.4. *When $m = 2$, then*

$$\Gamma^2_{g^*} = \Gamma^2_{g^*} - 2\gamma_2 - \gamma_1 + 2\gamma_3, \tag{5.21}$$

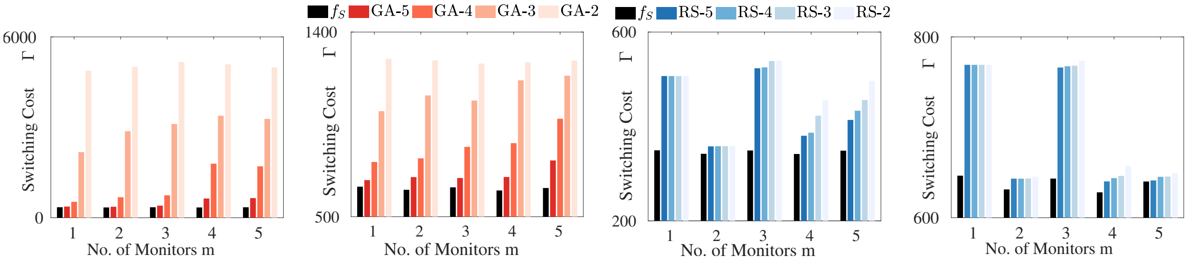
$$\Gamma^2_{f^*} = \Gamma^2_{f^*} - (2D-1)\gamma_2 + \gamma_3 + (\frac{H}{2}\gamma_1 + \gamma_2)e(L), \tag{5.22}$$

where $e(L) = 1$ if L is even, and $e(L) = 0$ otherwise. The minimum switching cost satisfies $\Gamma^2_{min} \leq \min\left\{\Gamma^2_{f^*}, \Gamma^2_{g^*}\right\}$. Further if $m = 2k$, $k \geq 2$, then $\Gamma^m_{min} \leq \Gamma^2_{min} - (m-2)\gamma_1$.

Proof. Eq. (5.21) is achieved by evenly dividing the space into two parts, and augment the divided path segment of $f^*_{m=1}$ and $g^*_{m=1}$ by adding extra Type-3 edges, as shown in Figure 5.7c and 5.7d. For example, in $f^*_{m=2}$, a Type-3 edge (black arrow in Figure 5.7c is added for each row. When L is even, the middle line of Type-1 edges needs to be divided, so a longer edge (solid yellow arrow) is added. Since the cost metric γ is a linear combination of d_S and d_A, the cost of the solid yellow arrow is equal to the summation of costs ($\frac{H}{2}\gamma_1 + \gamma_2$) of the two dashed yellow arrows. Similarly for $g^*_{m=2}$ (Figure 5.7d), two Type-2 edges and one Type-1 edges are removed, while one Type-3 edges are added. Then the minimum switching cost Γ^2_{min} is upper bounded by the minimum of the two. For $m = 2k \geq 4$, applying Lemma 5.3 to Γ^2_{min} yields the result. □

Combining Lemma 5.2, 5.3 and 5.4, we have the upper-bounds of the total switching cost Γ_f in Theorem 5.1. □

As a constructive proof, Theorem 5.1 also sketches the proposed deterministic strategy (denoted as f_S). To validate these results (strategy f_S and upper bounds), we conduct numerical simulation, whose configuration is enumerated in Table 5.4. Note that the width of the spectrum block $\mathcal{S}$ does not have any unit (similarly for monitoring power parameter δ). We eliminate the unit, instead of plugging in parameters of real-world hardware *e.g.*, [91, 84], because there will



(a) GA $\beta_S = 0.1$, $\beta_A = 0.9$ **(b)** GA $\beta_S = 0.8$, $\beta_A = 0.2$ **(c)** RS $\beta_S = 0.1$, $\beta_A = 0.9$ **(d)** RS $\beta_S = 0.8$, $\beta_A = 0.2$

Figure 5.8 The proposed strategy (black on the far left) achieves a lower switching cost, compared to the genetic algorithm solution (red bars in (a-b)) and that the greedy-based random search solution (blue in (c-d)), in different switching cost coefficients (β_A and β_S) settings.

be a (mere) change of constant when different units are applied, which is insignificant (and more confusing) in validating the efficiency of the proposed deterministic strategies.

Table 5.4 Simulation configuration for coverage time of strategy f_S^m.

Parameter	Description	Value
$n = \lvert\mathcal{V}\rvert$	number of assignment points	394
$m = \lvert\mathcal{M}\rvert$	number of monitors	$[1, 5]$
$\mathcal{A}$	the geographical region of interest	$[0, 10]^2$
$\lvert\mathcal{S}\rvert$	width of spectrum block	18
ϵ	scaling coefficient in distance d_{SA}	5
δ	radius of the monitoring power	$\frac{\sqrt{5}}{2}$
(L, D, H)	tessellation parameters (see Table 5.3)	(5,5,10)
β_S, β_A	switching cost coefficients (see Definition 5.7)	$(0, 1)$

The proposed strategy (black bars on the far left in Figure 5.8, labeled as f_S) is compared with the best solutions found by genetic algorithm (red-toned bars in Figure 5.8a-b, labeled as GA-x, which means 10^x iterations are executed to obtain this result), which is a commonly used heuristic for MTSP, and those found by a greedy-based random search (blue-toned bars in Figure 5.8c-d, labeled as RS-x, which means 10^x iterations are executed to obtain this result). The proposed f_S achieves a very close switching cost to that of the best solution/strategy output by GA after more than 10,000 iterations, under different switching cost coefficient settings. In fact, when m is small, the best solution provided by GA bears great resemblance to the proposed strategy, in terms of traversed edge types, as well as the breaking points (*i.e.*, the way to divide the optimal path of $m = 1$). Similar observations can be obtained in the comparisons with random search, despite that the optimal solution can not be easily found by the random search due to its greedy nature.

5.4.2 Detection Time of the Deterministic Strategy f_S

Though deterministic strategy f_S proposed in Theorem 5.1 proves to be a good strategy in terms of coverage time ($T_S^m = \lceil \frac{|V|}{m} \rceil$) and switching cost, its detection performance is not satisfying. In fact, it suffers from a 'wandering hole' problem when adversarial spectrum culprits are present. So in this subsection, we discuss the detection performance of the deterministic strategy f_S.

5.4.2.1 Detecting Persistent Culprits R_p

The expected detection time $\mathbb{E}(\tau_p(f_S))$ of different persistent culprits R_p under strategy f_S can be found in Table 5.2, given that the initial position of R_p (at time $t = 0$) is chosen randomly from the assignment space V. As can be seen, the expected detection time $\mathbb{E}(\tau_p(f_S))$ are bounded above by the coverage time $T_S^m = \lceil \frac{n}{m} \rceil$. Proofs of these results are listed as follows.

Proof. **(Expected detection time presented in Table 5.2.)**

In general, for a strategy $\{f_t^m\}_{t \in \mathcal{T}}$ conducted by a set $\mathcal{M}$ of monitors with $q(\delta)$-monitoring power, the probability that a culprit R_p is detected in time step $t = k$ is $p_k := q \sum_{v \in f_k(\mathcal{M})} g_{R_p}(v)$. Since R_p is persistent, *i.e.*, its exploiting pattern does not change over time, the probability that it not detected until time step k can be derived as $\mathbb{P}\big(\tau_{R_p}(f_t) = k\big) = p_k \prod_{i=0}^{k-1}(1-p_i)$, then the expected detection time of a persistent culprit R_p with PMF $g_{R_p}(v)$ can be obtained by

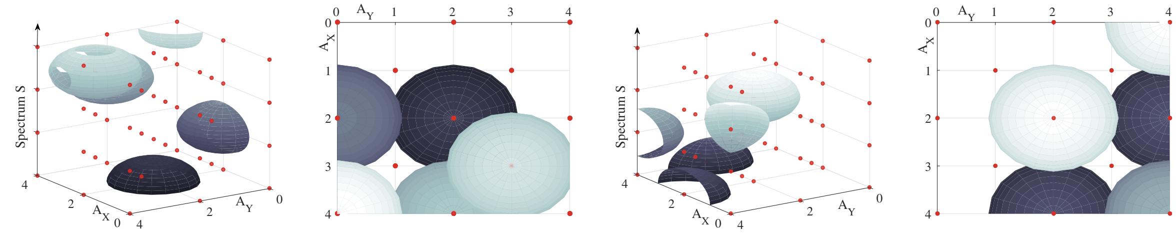
$$\mathbb{E}\big(\tau_p(f)\big) = \sum_{k=1}^{\infty} k p_k \prod_{i=1}^{k-1}(1-p_i), \tag{5.23}$$

where $p_0 = 0$ is added to keep its form consistent. Next, we discuss the detection time of the three types of persistent spectrum culprits.

Case 1. Stationary R_s without DSA capability, who chooses $x \in X$ (or equivalently $v \in V$) uniformly at random, and stays there for any $t > 0$. Therefore, the probability that this culprit is detect at time step k is $\mathbb{P}(\tau_{R_s}(f_S) = k) = \frac{1}{T_S}$, indicating a uniform distribution for $k = 1, 2, \ldots, _S$, and $\mathbb{P}(\tau_R(f_S) > T_S) = 0$. The expected detection time $\mathbb{E}(\tau_{R_s}(f_S)) = \frac{T_S}{2}$ is bounded by the coverage time of the deterministic strategy f_S.

Case 2. Stationary R_{sd} with DSA capability, that fixes its location $a \in \mathcal{A}$, and hops to any frequency chosen uniformly at random from $\mathcal{S}$, resulting in a $R_{sd}(t)$ such that $p_a(R_{sd}(t)) = a$ for all $t > 0$. Exact analysis of $\mathbb{E}(\tau_{R_{sd}}(f_S))$ is fairly difficult, due to the slight overlap of monitoring powers at location a. But for every time slot $t = k$, the probability that culprit R_{sd} is identified, is strictly less than that the more capable R_{md} is identified, *i.e.* $\mathbb{P}(\tau_{R_{sd}}(f_S) = k) < \mathbb{P}(\tau_{R_{md}}(f_S) = k)$. Therefore, $\mathbb{E}\left(\tau_{R_{sd}}(f_S)\right) \leq \mathbb{E}\left(\tau_{R_{md}}(f_S)\right)$.

Case 3. Mobile R_{md} with DSA capability, whose exploit sequence is randomly chosen from X (or equivalently discrete space V) during every time step. Since there are m monitors in the assignment space V, during every time step t, the probability that the culprit R_{md} is not detected is $1 - p = \frac{n-m}{n}$. Consequently r.v. $\tau_R(f_S)$ is geometrically distributed with parameter



(a) $t = 1$ side view **(b)** $t = 1$ top view **(c)** $t = 2$ side view **(d)** $t = 2$ top view

Figure 5.9 Illustration of a *wandering hole*: Five monitors are deployed in region $\mathcal{A} = [0,4]^2$ with $\delta = \frac{\sqrt{5}}{2}$. Their coverage C_t at time t, is the enclosed space of the blue (partial) balls and the boundary, while the outter space corresponds to the 'hole', that is 'wandering' (changing) in X over time.

p, that is, $\mathbb{P}(\tau_R(f_S) = k) = (1-p)^{k-1}p$. Hence its expected value $\mathbb{E}\left(\tau_R(f_S)\right) = \frac{n}{m} \simeq T_S$. □

5.4.2.2 Detecting an Adversary Culprit R_a

To maintain a low switching cost, the deterministic strategy f_S will be repeated (in forward/reverse direction in odd/even cycles) after $T_S = \lceil \frac{n}{m} \rceil$ time. Consequently it is possible for culprits with learning capabilities to observe the deployment pattern of monitors, and even *predict* where monitors will not be (*e.g.*, the coverage hole shown in Figure 5.3b) in the next time step. Then culprits can continue chasing the 'hole' for an infinitely long time, as if hiding in a 'wandering hole' of the dynamically changing coverage $C_t(f_{S,t})$.

Definition 5.8. *Strategy* $\{f_t\}_{t\in[1,T]}$ *is said to suffer from a* ***wandering hole*** *problem, if an adversarial culprit can exploit the system for infinitely long time, i.e.,* $\mathbb{E}\big(\tau_a(f_t)\big) = \infty$.

An example of the wandering hole problem is shown in Figure 5.9, where activities in spectrum block $\mathcal{S}$ over region $\mathcal{A} = [0,4]^2$ are being monitored by $m = 5$ monitors. Red dots indicate the assignment points of V, calculated by the Kelvin structure tessellation (in Section 5.3), while space enclosed by shaded spheres corresponds to the monitoring power of the deployed monitors[14], whose tuning frequency is identified by the darkness of the shade. The white space outside of these spheres corresponds to spectrum slices and locations that a culprit can exploit without being detected, *i.e.* a spectrum *hole* in the monitoring coverage C_t. At time $t = 1$, consider an adversarial culprits R_a located at $(3,3)$ occupying lower frequency portion in the 'hole'. From previous observation, culprit R_a can easily find spectra-location points to exploit in the next time slot $t = 2$, because a deterministic monitoring strategy f_S repeats itself periodically, *i.e.*, Void(2) identifies the 'hole' exactly as the 3-D region in X where the probability for it to be monitored during $t = 2$ equals to zero. Consequently, R_a can safely stay at the current location, and continue occupying the current spectrum slice without being detected. In other words, adversarial R_a can swiftly hide in the 'wandering hole' indefinitely, unless the

[14] Since we are interested in the closed space $\mathcal{S} \times \mathcal{A}$, spheres indicating the boundary of monitoring power are trimmed, when they intersect with the boundary of space $X = \mathcal{S} \times \mathcal{A}$.

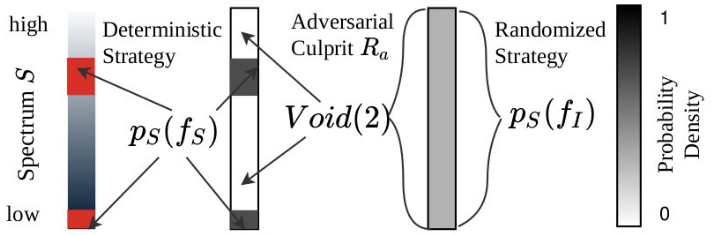


Figure 5.10 Root cause of the *wandering hole* problem: difference in visiting probability (density).

deterministic deployment strategy f_S changes. In fact, the wandering hole problem exists in any deterministic strategy. Once the SAS strategy (or more precisely, its probability distribution) is known by an adversarial culprit R_a, this prior knowledge can be leveraged by R_a to actively dodge monitors, whenever there is a hole in the current coverage, *i.e.*, $X \setminus C_t \neq \phi$.

5.5 Patching the 'Wandering Hole': Randomized Strategies

The wandering hole problem exposes a defect of deterministic strategies against adversarial culprits R_a. The root cause of this defect is that, taking advantage of its prior knowledge, *i.e.*, the *difference* among visiting probabilities on different assignment points, an adversarial culprit R_a is able to determine the spectrum 'hole' to exploit, *i.e.*, Void(t). The sharper the difference, the clearer the boundary of the 'hole', and the larger the chance for the culprit R_a to dodge monitors in the next time step. For instance, under the deterministic strategy f_S shown in Figure 5.9, a culprit R_a located at $a = (3, 2)$ can easily identify Void(2) $\subset \mathcal{S} \times \{a\}$ due to the prominent difference in probability (the second 'bar' from the left in Figure 5.10).

5.5.0.1 Motivation and Intuition

Knowing the root cause of the 'wandering hole' problem, a straight-forward countermeasure is to better protect, or frequently change the SAS strategy, such that obtaining visiting probabilities, or the strategy implementation, are more difficult to culprits, which requires constant effort in the deployment stage. Nonetheless, we can achieve the same goal if we carefully design a SAS strategy, from which no useful 'knowledge' can be derived, even if it is known to the culprit. In other words, we can fully *randomize* the deployment, such that every assignment point is visited with the same probability in the long run. Consequently, there is no probability difference, and hence no boundary of spectrum holes for culprits to locate, as illustrated by the (uniform) grey bar, second from the right in Figure 5.10. In addition, monitors follow no pattern at all when switching to a different assignment point, so the adversarial culprits essentially become persistent ones, in the sense that their exploit patterns become the same. Therefore, such randomized strategies will be effective in culprit detection.

On the other hand, as we shift focus from low-cost sweep-coverage to quick culprit detection, we switch gear from switching cost (Definition 5.7) to switching capacity (Definition

5.2), the latter of which can be viewed as a binary quantification of the former against a cost threshold. As a result, the underlying graph (G_M, G_R) (defined in Section 5.3.2.3) changes from complete graphs with numerical edge weights (cost), to sparser simple graphs without edge weights, pruned according to the switching capacity of monitors and culprits. This change is also consistent with the SAS scenario change (from dedicated to crowd-source[15]): considering a switching action is actually a change of surrogate monitors, randomized strategies are natural for the crowd-source monitoring scenario, where the switching cost does not scale with distance d_{SA}. To be more specific, with a centralized coordinator or guaranteed communication among participants, switching between surrogate monitors can be timely coordinated, so the entire assignment space V is contained in the switching capacity of all monitors, which translates to the unlimited switching capacity case, *i.e.*, $\alpha_M = \infty$; in case of distributed control, when change of surrogate monitors needs to be completed with local communication, the switching capacity of monitors will be upper-bounded, *i.e.*, $\alpha_M < \infty$, due to their limited communication ranges.

Next, we introduce two simple, but effective randomized strategies.

5.5.1 Randomized SAS Strategies

We consider two randomized strategies that requires different levels of coordination and switching capacities: the independent I-strategy f_I and the distributed D-strategy f_D.

(1) *I-strategy* f_I. During each time step, each monitor $M_i \in \mathcal{M}$ switches to point $v_i \in V$ uniformly at random, and independently of others, within its switching capacity α_M. The surveillance process is then equivalent to a composite random walk of $m = |\mathcal{M}|$ walkers, each independently generating a sequence $\{f^m_{t,I}(M)\}_{t\in\mathcal{T}}$, on the monitoring subgraph G_M. When G_M is (close to) regular, *i.e.*, the number of assignment points reachable in one switching action is almost the same starting from every point in V, the uniform transition probability leads to a convergence-guaranteed distribution of visiting probabilities, that is, $\pi_v = \frac{1}{n}$, $\forall v \in V$.

(2) *D-strategy* f_D. Assignment space V is evenly divided into m disjoint subsets $\{V_i\}_{M_i\in\mathcal{M}}$, such that points in each subset V_i are within monitors' switching capacity α_M, and sets in collection $\{V_i\}_{M_i\in\mathcal{M}}$ compose a partition of V. During time step t, each monitor $M_i \in \mathcal{M}$ switches to point $f^m_{t,D}(M_i)$, chosen uniformly at random from its own subset V_i, which contains $n_m = \lceil \frac{n}{m} \rceil$ assignment points. Thus, the D-strategy is equivalent to m independent single-walker random walks, each on a smaller complete graph K_{n_m}. Moreover, graph K_{n_m} is regular, so the stationary visiting probabilities are also uniform.

Based on this design, we first discuss the basic case of $\alpha_M = \alpha_R = \infty$, *i.e.*, SAS strategy without switching constraint, in this section, and leave the more complicated $\alpha_M < \infty$ case for the next section. As we will show, both the coverage time and detection time of the two randomized strategies: i) are bounded, indicating their efficacy; and ii) scale as $O(\frac{1}{m})$ with respect to the number of monitors m, revealing their efficiency.

[15] For SAS with dedicated monitors, this modeling approach with switching capacity is also valid, if we set a fixed switching cost threshold γ^*, and determines switching capacity $\alpha_M < \infty$ according to this threshold.

5.5.2 Coverage Time of the Two Randomized Strategies f_I and f_D

Because of the unlimited switching capacity of both monitors and culprits, the underlying monitoring subgraph G_M are complete graphs, and both the I- and D-strategy are equivalent to random walks, on K_n and K_{n_m}, respectively. The coverage time T_I and T_D become well-defined r.v.'s that take value in $[1, \infty)$, and their expected value $\mathbb{E}(T_*)$ are referred to as the *cover time* [7]. This quantity is well-studied for a single walker case, which correspond to the single monitor case (I-strategy and D-strategy are exactly the same, *i.e.*, $f_I^1 = f_D^1$), but considering that it is helpful for the latter cases, we present it in Lemma 5.5 for completeness reasons.

Lemma 5.5. *The expected coverage time of strategy f_I^1 (or equivalently f_D^1) on the assignment space V is $\mathbb{E}(T_I^1) = n\mathcal{H}_n$, where $\mathcal{H}_n = \sum_{i=1}^n \frac{1}{i}$ denotes the n-th Harmonic number, and $n = |V|$ is the total number of assignment points.*

Proof. The surveillance sequence generated by strategy f_I^1 (f_D^1) is a single-walker random walk on K_n. Let r.v. $T_i \geq 0$ denote the time interval between the first hitting time of the $i-1$-th vertex and that of the i-th vertex, for any $i \geq 2$. Then T_i is geometric distributed with $p_i = \frac{n-i+1}{n}$, and $T_1 = 1$. So the expected coverage time can be calculated as $\mathbb{E}(T_I^1) = \mathbb{E}\left(\sum_{i=1}^n T_i\right) = \sum_{i=1}^n \frac{n}{n-i+1} = n\mathcal{H}_n$. □

Next, we discuss the case of multiple monitors for the I- and D-strategy, respectively.

5.5.2.1 Coverage Time of the I-strategy T_I^m

For an I-strategy f_I^m carried out by m monitors, the expected coverage time $\mathbb{E}(T_I^m)$ can be bounded by the following theorem.

Theorem 5.2. *For a set of $m = |\mathcal{M}|$ monitors that follow the I-strategy $\{f_t^m\}_{t\in\mathcal{T}}$ in the assignment space V, the expected coverage time is upper bounded by*

$$\mathbb{E}(T_I^m) \leq e(n-1)\left[0.562 + 0.768\frac{\mathcal{H}_n}{m}\right], \tag{5.24}$$

where $n = |V|$ is the number of assignment points in V.

Proof. When $m = 1$, the expected coverage time $\mathbb{E}(T_I^1) = n\mathcal{H}_n$ from Lemma 5.5. For $m > 1$, let $T_{I,i}^1$ denote the coverage time of a single monitor $M_i \in \mathcal{M}$. Then $\{T_{I,i}^1\}_{i=1}^m$ is a set of i.i.d. random variables with $T_{I,i}^1 \overset{d}{=} T_I^1$ for any monitor $M_i \in \mathcal{M}$. So

$$\mathbb{E}(T_I^m) \leq \mathbb{E}(\min_{1\leq i\leq m} T_{I,i}^1) \leq \mathbb{E}(T_{I,i}^1) = \mathbb{E}(T_I^1). \tag{5.25}$$

Let $H(x, y) := \min_{t>0}\{f_t^m(M_i) = y \mid f_0^m(M_i) = x\}$ denote the first hitting time of monitor $M_i \in \mathcal{M}$ on assignment point $y \in V$, given that M_i started its walk (surveillance) from $x \in V$. Since monitor M_i is inter-changeable with M_j, the notation of monitor (i) can be suppressed in the hitting time. Each monitor walks independently on the complete graph K_n, so $\mathbb{E}(H(x,y)) = n-1$, for any point $x, y \in V$.

Suppose at time $t = 0$ (one step ahead of the SAS process starts), all monitors are assigned to the same point x, that is, $f_0^m(M_i) = x$ for all $M_i \in \mathcal{M}$. Then for a fixed point $u \in V$, the probability that a random walk of length $e(n-1)$ does not hit u is upper bounded by the Markov Inequality [7]:

$$\mathbb{P}\big(H(x,u) > e(n-1)\big) \overset{Markov}{\leq} \frac{E(H(x,u))}{e(n-1)} = e^{-1}. \tag{5.26}$$

Then for any integer $r > 1$, probability $\mathbb{P}\big(H(x,u) > er(n-1)\big) \leq e^{-r}$, since the entire walk can be viewed as r individual trails trying to hit u simultaneously. Therefore, the probability that m independent (single-walker) random walks all starting from x, have not hit point u up to time $t = er(n-1)$, can be upper-bounded by

$$\mathbb{P}\Big(\min_{M_i \in \mathcal{M}} \{H_i(x,u)\} > er(n-1)\Big) \leq e^{-mr}. \tag{5.27}$$

Let $r = \lceil \frac{\ln n + \gamma}{m} \rceil$, where $\gamma = \lim_{n\to\infty}(\mathcal{H}_n - \ln n)$ is the Euler-Mascheroni constant. Then the probability that the m random walkers have not covered all points in V by time $er(n-1)$, or equivalently the coverage time T_I^m is greater than $er(n-1)$ can also be upper bounded:

$$\mathbb{P}\big(T_I^m > er(n-1)\big) \leq e^{-mr} \leq e^{-\gamma}. \tag{5.28}$$

Also, notice that expected values $\mathbb{E}(T_I^m) \leq \mathbb{E}(T_I^1) = n\mathcal{H}_n$ from Eq. (5.25). Therefore, the expected coverage time can be obtained as

$$\begin{aligned} \mathbb{E}(T_I^m) &\leq er(n-1)\cdot(1-e^{-\gamma}) + \mathbb{E}(T_I^1)\cdot e^{-\gamma} \\ &\leq \frac{n-1}{m}\left[(\ln n + \gamma + m)(e - e^{1-\gamma}) + \mathcal{H}_n e^{-\gamma}\right] \\ &\leq \frac{e(n-1)}{m}\left[(\mathcal{H}_n + m)(1-e^{-\gamma}) + \mathcal{H}_n e^{-(1+\gamma)}\right], \end{aligned} \tag{5.29}$$

and plugging in values of γ and e yields the result. □

Though not a tight bound, Theorem 5.2 reveals the scaling law of the expected coverage time with respect to size n of space V, and number of monitors m, that is, $\mathbb{E}(T_I^m) = O(\frac{n \ln n}{m})$.

5.5.2.2 Coverage Time of the D-strategy T_D^m

The expected coverage time of D-strategy f_D^m can also be bounded above.

Theorem 5.3. *For a set of m monitors following the D-strategy f_D^m on the assignment space*

V of size n, the expected coverage time $\mathbb{E}(T_D^m)$ is upper bounded by

$$\mathbb{E}(T_D^m) \leq n_m \mathcal{H}_{n_m} + \frac{n_m\sqrt{m-1}}{2(n_m-1)} \left[7(n_m)^2 - 11n_m + 2\right]^{\frac{1}{2}}, \tag{5.30}$$

where $n_m = \lceil \frac{n}{m} \rceil$.

Proof. Let r.v. T_{M_i} denote the time when monitor M_i has covered every assignment point in its own subset $V_i \subset V$ for the first time, where $|V_i| \overset{\triangle}{=} n_m = \lceil \frac{n}{m} \rceil$, and we refer to T_{M_i} as the sub-coverage time. Under the D-strategy f_D^m, the sequence of assignment points to be scanned by monitor M_i, can be viewed as a trail generated by a random walker on the complete graph K_{n_m} of size n_m. Therefore, by Lemma 5.5, the expected sub-coverage time $\mathbb{E}(T_{M_i})$ by monitor $M_i \in \mathcal{M}$ can be obtained as $\mathbb{E}(T_{M_i}) = n_m$.

In addition to the mean, we know the distribution of every T_{M_i}, particularly its mean and variance, from the proof of Lemma 5.5, because r.v. $T_{M_i} = \sum_{k=1}^{n_m-1} T_k$, where each T_k is geometrically distributed with parameter $\frac{n_m-k+1}{n_m}$. Hence $\mathbb{E}(T_k) = \frac{n_m}{n_m-k+1}$ and $Var(T_k) = \frac{n_m(k-1)}{(n_m-k+1)^2}$. Then the variance of r.v. T_{M_i} can be derived as

$$\begin{aligned} Var(T_{M_i}) &= \sum_{k=1}^{n_m} Var(T_k) = \sum_{k=1}^{n_m} \frac{n_m(k-1)}{(n_m-k+1)^2} \\ &\leq (n_m)^2 \left(1 + \sum_{j=2}^{n_m} \frac{1}{j^2-1}\right) = (n_m)^2 \left(\frac{7}{4} - \frac{2n_m+1}{2(n_m-1)n_m}\right). \end{aligned} \tag{5.31}$$

The coverage time of strategy f_D^m, *i.e.*, the first time when every points in V is sweep-scanned, actually equals to the time when the last monitor, say M_j, finishes scanning the last point in its own subset V_j. Consequently, T_D^m equals to the maximum of m i.i.d. r.v.s, that is,

$$T_D^m = \max_{M_i \in \mathcal{M}} \{T_{M_i}\}, \tag{5.32}$$

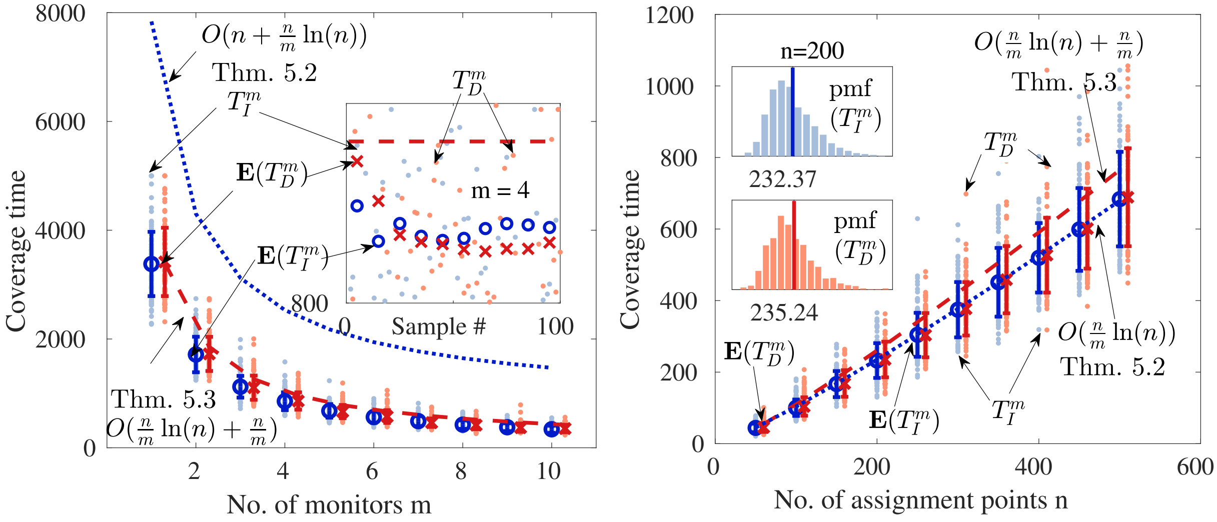
which can be upper-bounded using the technique in [15, Eq.(3)],

$$\begin{aligned} \mathbb{E}(T_D) &\leq \max_{M_i \in \mathcal{M}} \{\mathbb{E}(T_{M_i})\} + \sqrt{(m-1)Var(T_{M_i})} \\ &= \mathbb{E}(T_{M_i}) + \sqrt{(m-1)Var(T_{M_i})}. \end{aligned} \tag{5.33}$$

Plugging Eq. (5.31) into Eq. (5.33) yields the upper-bound. □

5.5.2.3 Numerical Validation

Figure 5.11 shows the expected coverage time of I-strategy ($\mathbb{E}(T_I^m)$, blue '○' markers) and D-strategy ($\mathbb{E}(T_D^m)$, red '×' markers) with respect to $m \in [1, 10]$, number of monitors, and $n = |V| \in [50, 500]$, size of the assignment space, respectively. Numerical results of T_I and



(a) Scaling as $O(\frac{1}{m})$ over $m = |\mathcal{M}|$ with $n = 500$. **(b)** Scaling as $O(n \ln n)$ over $n = |V|$ with $m = 5$.

Figure 5.11 The expected coverage time of both the I- ($\mathbb{E}(T_I)$) and D-strategy ($\mathbb{E}(T_D)$) are $O(\frac{n}{m}\ln n)$, as predicted by Theorem 5.2 and Theorem 5.3, respectively.

T_D are shown in dots, while their mean and standard deviation are shown with markers and bracketed bars. The case of four monitors ($m = 4$) is zoomed in the inner box of Figure 5.11a, and the distribution of coverage time (for $n = 200$ assignment points) is shown in the inner box of of Figure 5.11b. We draw the following observations from the numerical validation: i) Theorem 5.3 (red dashed line) is a tight bound on the coverage time of D-strategy when m is small; ii) Theorem 5.2 (blue dotted line), though not a tight bound, accurately describes its $O(\frac{n}{m}\ln n)$ scaling behavior; iii) I-strategy and D-strategy have very close coverage time performances, not only in the mean sense, but also in distribution, as shown in the inner boxes of Figure 5.11b, which implies that the more demanding I-strategy (in terms of level of coordination and switching capability of monitors) can be safely substituted by the distributed D-strategy, with the same guaranteed performance in sweep-coverage; and iv) both expected coverage times can be described as $O(\frac{n}{m}\ln n)$ (blue dotted line in Figure 5.11b), indicating that this scaling law can be used to predict the number of monitors needed to reach the coverage goal for SAS over a fixed spectrum range and geographical region, when the resolution of surveillance result is determined by the $q(\delta)$-monitoring power.

Note that unit of the coverage time is time step, whose length can be evaluated when parameters in Table 5.4 are determined in a real-world scenario, according to the monitoring power of SAS monitors. We do not incorporate specific units in the analysis or simulation, because our main objective is to find the scaling behavior of strategy performance with respect to the problem size (in this case n) and the amount of SAS resource (in this case m), such that generic design guidelines can be obtained for different SAS application scenarios. In addition, compared with the deterministic strategy f_S in Section 5.4, which can achieve a $\frac{n}{m}$ coverage time, randomized strategies seem to be at disadvantage in fulfilling the coverage goal, but as we will show in the following subsection, they are favorable in spectrum culprits detection.

5.5.3 Bounded Detection Time of Adversarial Culprits

As briefly analyzed in the example (Figure 5.10), the advantage of adversarial culprits over deterministic strategies is lost, when facing monitors running randomized strategies f_I and f_D, because their prior knowledge (visiting probabilities) is compromised by the uniform probability distribution in f_I and f_D. Consequently, randomized strategies do not suffer from the 'wandering hole' problem, that is, the detection time is bounded.

Theorem 5.4. *Under strategy f_I^m and f_D^m, the expected detection time $\mathbb{E}(\tau_R(f_*))$ of an adversarial culprit R_a is upper-bounded, if the detection probability is strictly positive, i.e., $q \geq q^* > 0$.*

Proof. Consider m monitors over the assignment space V of size $n = |V|$, each has $q(\delta)$-monitoring power. That is to say, if the spectra-location distance between a monitor M_i and a culprit R_a is less than δ, the probability that this particular culprit R_a will be detected by M_i during one time step with probability $q \geq q^*$. To I- and D-strategy, an adversarial culprit R_a is equivalent to a mobile persistent culprit R_{md} (see Table 5.2) in terms of detection time, because from the perspective of R_a, the probability of any assignment point being visited during the next time step is the same.

First we discuss the I-strategy f_I^m. The detection time is actually the first meeting time between the culprit and any of the m monitors, while both the culprit and monitors randomly walk on the complete graph K_n. By Eq. (5.23), detection time $\tau_R(f_I^m)$ is geometrically distributed with parameter $p_I = 1 - (1 - \frac{q}{n})^m$, which equals to the probability that R is caught by at least one of the m monitors in a single time step. Therefore,

$$\mathbb{E}\big(\tau_a(f_I^m)\big) = \frac{1}{p_I} = \Big[1 - (1 - \frac{q}{n})^m\Big]^{-1}. \tag{5.34}$$

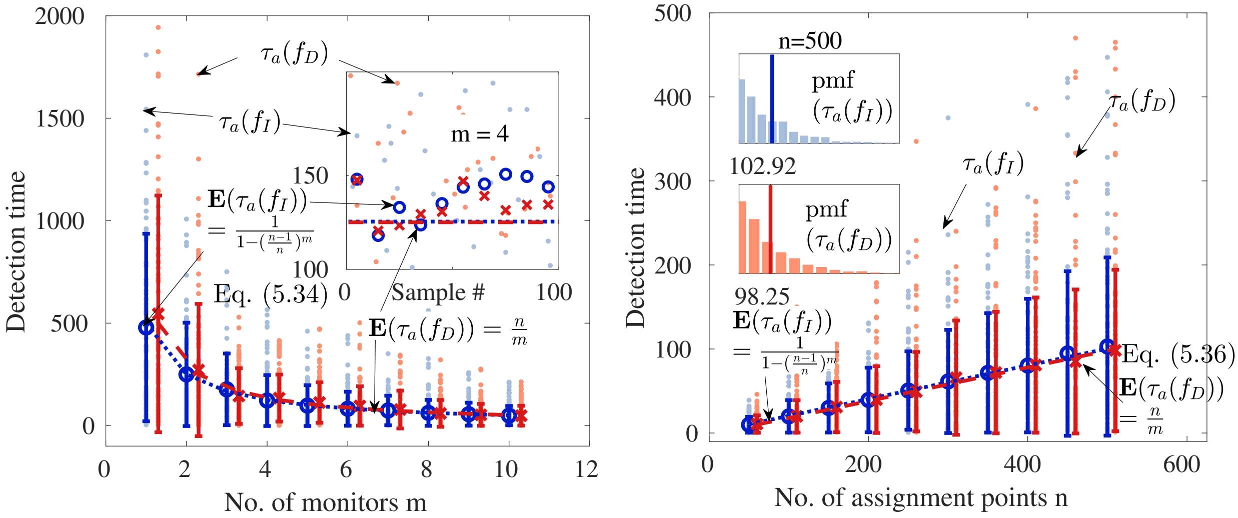
Then given that the detecting probability/reliability q is lower-bounded by a positive constant $q^* > 0$, Eq. (5.34) can be upper-bounded by

$$\mathbb{E}\big(\tau_a(f_I^m)\big) \leq \frac{1}{1 - \Big[(1 - \frac{q^*}{n})^{\frac{n}{q^*}}\Big]^{\frac{q^* m}{n}}} \leq \frac{1}{1 - e^{-\frac{q^* m}{n}}}. \tag{5.35}$$

For the D-strategy f_D^m, each monitor switches independently, and occupies a different point in V during each time step. Therefore, the probability that culprit R_a is detected by any of the m monitors during t is $p_D = q\frac{m}{n}$, and the corresponding expected detection time is also upper-bounded:

$$\mathbb{E}(\tau_a(f_D^m)) = \frac{1}{p_D} = \frac{n}{qm} \leq \frac{n}{q^* m}. \tag{5.36}$$

Therefore, any adversarial culprit R_a will be detected in finite time, given that detection probability $q \geq q^* > 0$. □



(a) Scaling as $O(\frac{1}{m})$ over $m = |\mathcal{M}|$ with $n = 500$.

(b) Scaling as $O(n)$ over $n = |V|$ with $m = 5$.

Figure 5.12 The expected detection time of an adversarial culprit in the assignment space of size n is $O(\frac{n}{m})$, under both the I and the D-strategy with full reliabiltiy $q = 1$.

Theoretically, it is possible that detecting probability q is minimal, due to the large radius δ in the $q(\delta)$-monitoring power, so the resulting detection time $\mathbb{E}(\tau_a(f_I^m))$ and $\mathbb{E}(\tau_a(f_D^m))$ tend to infinity. However, this can be easily fixed if the radius parameter δ is adjusted in the space-tessellation step, so that q is boosted to an acceptable level. Consequently, we conclude that randomized monitoring strategies (f_I and f_D) do not suffer from the 'wandering hole' problem.

To validate the advantage of randomized strategies against adversarial culprit, *i.e.*, Theorem 5.4, culprit detection is simulated under the same spectra-location space setting as the sweep-coverage validation. Detection time samples are shown as light-blue (I-strategy) and light-red (D-strategy) dots in Figure 5.12, which corresponds to the $q = 1$ case (perfect detection), and Figure 5.13, which corresponds to the imperfect detection ($q = 0.8 < 1$) case. We highlight the following observations. i) Not only are the detection time of I-strategy and D-strategy bounded (and hence no 'wandering hole' problem), their expectations can be accurately calculated with Eq. (5.34) and (5.36), once the number of monitors m and size of the assignment space n are fixed. ii) Bounds in Eq. (5.34) and (5.36) hold for the imperfect detection case, as shown in Figure 5.13. iii) The detection performance of the I- and D-strategy are fairly close, which indicates that D-strategy can be a good distributed alternative to the I-strategy.

In addition, the $O(\frac{1}{m})$ scaling behavior in both coverage (Figure 5.11a) and detection (Figure 5.12a) time, indicates a linear 'speed-up' in SAS performance, when multiple monitors are employed in the randomized strategies. This behavior implies, as same as in the deterministic strategies, increasing the number of monitors (m) is an efficient performance-boosting measure. In addition, the bounds on detection time (or rather, accurate results in Eq. (5.34) and (5.36)) add to the predictability of randomized strategies, which can be fairly useful in the design stage of a SAS system, *e.g.*, estimating the number of monitors needed for culprit detection.

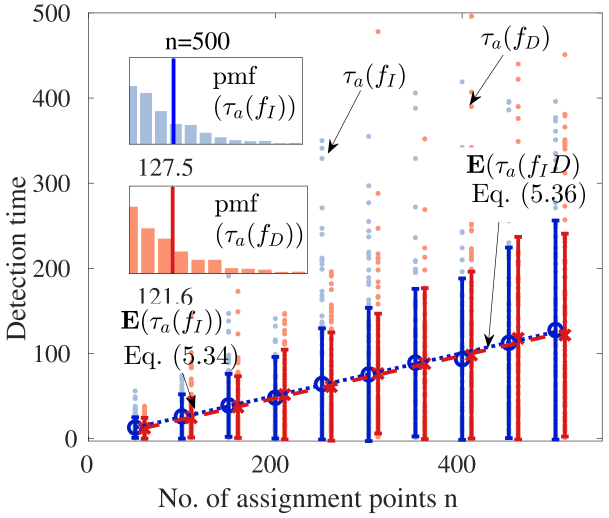


Figure 5.13 Expected detection time of culprits under imperfect detection (reliability $q = 0.8$) is also well captured by Eq. (5.34) and (5.36).

5.6 SAS with Limited Switching Capacities

In this section, we discuss the SAS process of m independent α_M-monitors (with detection probability $q = 1$) and an α_R-culprit on the assignment space V, in which the switching capacity of the monitors (α_M) and the culprit (α_R) are limited, *i.e.*, finite. Constrained switching capacity ($\alpha_M < \infty$) applies to the fully-distributed crowd-source SAS scenario, in which a switching is only possible if the two devices are within each other's communication range. Such a capacity limit can also be viewed as a binary quantification of the switching cost, in the sense that it cuts off any edge (v_i, v_j) (in a complete graph), whose associated switching cost $\gamma(v_i, v_j)$ exceeds a threshold, to obtain the monitoring/exploiting subgraphs. Analysis of a SAS process under switching capacity constraint, or as we formulate it, a composite graph walk process on graph (G_M, G_R) that is not a complete graph, is challenging due to the following reasons: i) theoretic analysis of random walks on general graphs is difficult, if at all possible, because existing mathematical tools are developed for graphs with special structures, *e.g.*, the complete graph [5] and the expander graph [31]; ii) monitors and culprits switch/walk on different graphs when $\alpha_R \neq \alpha_M$, which is not addressed by existing research on graph walk, *e.g.*, [32].

For the ease of discussion, we introduce *degree* $r_*()$ of an assignment point in the in the monitoring subgraph G_M and the exploiting subgraph G_R. Let $r_{\alpha_M}(v)$ denote the degree of point $v \in V$ in G_M, under constraint α_M, and $r_{\alpha_R}(v)$ denote the degree of v in G_R, under constraint α_R. Graph G_M and G_R are both subgraphs of a complete graph (degree $r(v) = n-1$ for every node v), which corresponds to the case of $\alpha_M = \alpha_R = \infty$ discussed in Section 5.5. When $\alpha_M = \alpha_R$, the culprit and the monitors can be viewed as walking on the same graph, *i.e.*, $G_M = G_R$. However, for cases when monitors are more 'powerful' than the culprit ($\alpha_M > \alpha_R$), edges in G_R are strictly sparser, *i.e.*, $E_R \subset E_M$, and vise versa.

5.6.1 Coverage and Detection Time through Regular Graph Approximation

Observe that assignment points (cell centers in the Kevin structure) in space V are quite 'structured', as shown in Figure 5.4d. As a result, subgraph G_M (respectively G_R) (*e.g.*, the weaker monitors v.s. powerful culprit case shown in Figure 5.6b) that build upon it is also 'structured', in the sense that degrees of most vertices in G_M (G_R) are roughly the same, except for the few near the boundary. Denote $r_M = \frac{1}{n}\sum_{i=1}^{n} r_{\alpha_M}(v_i)$ as the average degree of the monitoring subgraph G_M, and r_R as that of the exploiting subgraph G_R. We first approximate G_M and G_R as r_M- and r_R-regular graphs[16] (G_{r_M} and G_{r_R}), respectively, on which mathematical tools [7, 5, 32] come in handy.

5.6.1.1 Coverage Time $T^m_{r_M}$

Let $T^m_{r_M}$ denote the coverage time of m independent monitors on r_M-regular graph G_{r_M}, to differentiate from the actual coverage time T^m_I on the original monitoring subgraph G_M. On regular graph G_{r_M}, asymptotic bounds for $\mathbb{E}(T^m_{r_M})$ have been studied by multiple researchers. Among these, Alon *et.al.* [7] proved $\mathbb{E}(T^m_{r_M}) \sim \Theta(\frac{n \ln n}{m})$, as $n \to \infty$; Cooper, Frieze and Radzik[32] provided a similar but more accurate asymptotic result for random regular graphs, when the number of random walkers is not large, *i.e.*, $m = o(\frac{n}{\ln^2 n})$. It is shown in [31] that a uniformly chosen $r-$regular ($r \geq 3$) graph G_r is 'nice' with high probability (tending to one as $n \to \infty$), such that the expected coverage time $T^m_{r_M}$ follows from [32, Thm. 2], that is,

$$\mathbb{E}(T^m_{r_M}) \sim \frac{r_M - 1}{r_M - 2}\frac{n \ln n}{m}. \tag{5.37}$$

Under an I-strategy, each monitor switches to any assignment point within its switching capacity uniformly at random, resulting in a uniform stationary distribution $\pi_v = \frac{1}{n}$ over the assignment space V. In other words, each assignment point is visited roughly the same number of times as time proceeds, so we say the spectra-location space X is 'evenly' covered. Nevertheless, if a certain region (subspace of X) needs special attention, *e.g.*, due to higher presence of misbehavior, the probability to switch to a target assignment point in the switching capacity can be adjusted, such that a desired (possibly non-uniform) stationary distribution over V can be achieved through well-articulated algorithms, *e.g.*, the Metropolis-Hasting algorithm.

5.6.1.2 Detection Time $\tau_R(r_R, r_M)$

Unlike coverage time, for which existing research leads to direct solution, there is no proper mathematical tool to directly address the culprit detection problem, in which the monitors and the culprit may walk on *different* graphs, due to their different switching capacity limits. So we address the weaker monitors vs. powerful culprit ($\alpha_M < \alpha_R$) case (and its reverse $\alpha_M > \alpha_R$)

[16] This average-degree-based approximation is reasonable, but also introduce a gap when determining SAS performance for both sweep-coverage and culprit detection, which is discussed in Section 5.6.2.

in this subsection.

Let $\tau_R(r_R, r_M)$ denote the detection time of a culprit R (walking on the r_R-regular graph G_{r_R}), by m-monitors (walking on the r_M-regular graph G_{r_M}). There are two possible cases:

Case 1. *Same switching capacity* ($r_R = r_M = r$). Monitors $\mathcal{M}$ and the culprit R have the same switching capacity, such that $r_R = r_M = r$, and $G_{r_R} = G_{r_m} = G_r$, *i.e.*, both monitors and culprit R walk on an r-regular graph G_r. Applying the predictor-and-prey model [32, Theorem 3], the expected detection time of culprit R can be asymptotically bounded, that is,

$$\mathbb{E}(\tau_R(r, r)) \sim \frac{r-1}{r-2} \cdot \frac{n}{m}. \tag{5.38}$$

Case 2. *Different switching capacities* ($r_R \neq r_M$). Monitors $\mathcal{M}$ and the culprit R walk on different graphs, *i.e.*, $r_R \neq r_M$ such that $G_{r_R} \neq G_{r_m}$. We obtain the upper bound of the detection time $\mathbb{E}\left(\tau_R(r_R, r_M)\right)$ by considering a composite random walk.

Proposition 5.2. *Let $K = \frac{(n-1)!}{(n-m-1)!}$. Under an I-strategy with m monitors, the expected detection time of a culprit on graph (G_{r_M}, G_{r_R}) is upper bounded, i.e.,*

$$\mathbb{E}\left(\tau_R(r_R, r_M)\right) \leq 1 + \frac{K}{n^m}(4K^2 - 1). \tag{5.39}$$

Proof. Without loss of generality, we number the monitors in $\mathcal{M}$ as $\{1, 2, \ldots, m\}$. Let $\bar{\mathcal{M}} = \mathcal{M} \cup \{m+1\}$ denote a set of walkers, where the first m walkers correspond to monitors, who walk on the monitoring subgraph G_M, and walker $m+1$ refers to the culprit R, who walks on the exploiting subgraph G_R.

Consider a (single-walker) random walk $\{Z_t\}_{t \in \mathcal{T}}$ on graph $H = (V_H, E_H)$, where each vertex $\vec{v}$ is a vector consisting $m+1$ elements, that is, $V_H = \{\vec{v} = (v_1, v_2, \ldots, v_m, v_{m+1}) \mid v_i \in V,\ \forall i \in \bar{\mathcal{M}}\}$. Edges of graph H are constructed as follows: edge $(\vec{v}, \vec{w}) \in E_H$ exists, if and only if: i) $(v_i, w_i) \in E_M,\ \forall i \in \mathcal{M}$; and ii) $(v_{m+1}, w_{m+1}) \in E_R$. Graph H is a regular graph, since for every vertex $\vec{v} \in V_H$, the number of neighbors, or the degree of vertex $\vec{v}$, $d_H(\vec{v}) = r_M^m \cdot r_R$, given that (G_M, G_R) is approximated by the regular composite graph (G_{r_M}, G_{r_R}).

Now the SAS process, or more specially, the culprit detection process, can be described by the single-walker random walk $\{Z_t\}_{t \in \mathcal{T}}$ on graph H, thanks to its construction. Let set

$$B := \{\vec{v} \in V_H \mid \exists\ i \in \mathcal{M}\ s.t.\ v_i = v_{m+1}\}$$

denote the set of vertices where the culprit (walker $m+1$) is detected by one of the m monitors (co-locates with one of the m walkers in $\mathcal{M}$). Then the detection time of culprit R, is the first hitting time of set B by walk Z_t, that is,

$$\tau_R^{\vec{v}}(r_R, r_M) = \min_{t \in \mathcal{T}}\{t \mid Z_t \in B\}, \tag{5.40}$$

provided that walk Z_t started from initial position $Z_1 = \vec{v}$. Since both the monitors $\mathcal{M}$ and culprit R starts uniformly at random in the assignment space V at time $t = 1$, the probability that walk Z_t starts from vertex $\vec{v}$ can be calculated by $\mathbb{P}(Z_1 = \vec{v}) = \frac{1}{|V_H|} = n^{-(m+1)}$. With respect to set B, there are two possible cases:

Case 1. If walk Z_t starts from set B, that is, $Z_1 \in B$, the culprit R is immediately identified. This event happens with probability

$$\mathbb{P}\left(\tau_R^{\vec{v}}(r_R, r_M) = 1\right) = \mathbb{P}(Z_1 \in B) = \frac{|B|}{|V_H|} = 1 - \frac{n!}{(n-m-1)!} = 1 - nq. \tag{5.41}$$

Case 2. Otherwise (with probability $1 - \mathbb{P}\left(\tau_R^{\vec{v}}(r_R, r_M) = 1\right)$), the initial position $Z_1 \in V_H \setminus B$, with probability nq. In this case, since H is a regular, the following inequality follows from [5, Proposition 6.16]:

$$\mathbb{E}\left(\tau_R^{\vec{v}}(r_R, r_M) \mid \vec{v} \notin B\right) \leq 4|V_H \setminus B|^2. \tag{5.42}$$

Then the upper bound in Eq. (5.39) can be obtained by combing the two cases. □

Proposition 5.2 holds for every n, but when n is large, it is not easy to calculate K and n^m in Eq. (5.39). For this case, we have the following scaling law on the expected detection time.

Corollary 5.1. *The expected detection time of a culprit for a SAS process under the I-strategy on graph (G_{r_R}, G_{r_M}) satisfies*

$$\mathbb{E}\left(\tau_R(r_R, r_M)\right) = \Theta(\frac{n}{m}), \tag{5.43}$$

where the exploiting subgraph G_{r_R} and the monitoring subgrpah G_{r_M} are r_R- and r_M-regular graphs, respectively.

Proof. Let $r_1 = \min\{r_R, r_M\}$ and $r_2 = \max\{r_R, r_M\}$, then

$$\mathbb{E}\left(\tau_R(r_2, r_2)\right) \leq \mathbb{E}\left(\tau_R(r_R, r_M)\right) \leq \mathbb{E}\left(\tau_R(r_1, r_1)\right). \tag{5.44}$$

Also, from results of detection time on regular graphs, we have $\mathbb{E}\left(\tau_R(r_2, r_2)\right) \sim \frac{r_2 - 1}{m(r_2 - 2)} n = \Omega(\frac{n}{m})$ and $\mathbb{E}\left(\tau_R(r_1, r_1)\right) \sim \frac{r_1 - 1}{m(r_1 - 2)} n = O(\frac{n}{m})$. Then it follows that $\mathbb{E}\left(\tau_R(r_R, r_M)\right) = \Theta(\frac{n}{m})$. □

Compared to the SAS scenarios with unlimited switching capacity ($\alpha_M = \infty$) discussed in Section 5.5, the scaling laws in this regular graph approximation (coverage time Eq. (5.37), detection time Eq. (5.38) and Eq. (5.43)) differ only by a degree-determining constant, which is less than or equal to 2. Consequently, we expect that, upon the imposed switching capacity limit, scaling laws of both performance metrics over m and n to remain the same compared to that under no switching capacity limit.

5.6.2 Gap between (G_M, G_R) and Approximation (G_{r_M}, G_{r_R})

For Eq. (5.37) and Eq. (5.38) to hold, a regular graph needs to be ‘nice’ [31, pp. 733]. It is also shown [31] that a large (n large) r-regular graph G_r randomly selected from all r-regular graphs $\mathcal{G}_r$, is *almost-Ramanujan* with high probability, that is, the largest eigenvalue $\lambda_0(G_r)$ and the second largest eigenvalue $\lambda_1(G_r)$ of graph G_r's adjacency matrix satisfy

$$\lambda_1(G_r) \leq 2\sqrt{\lambda_0(G_r) - 1} + \epsilon, \tag{5.45}$$

where the $\lambda_0(G_r) = r$, as G_r is r-regular.

However, Eq. (5.45) does not necessarily hold for the actual composite graph (G_R, G_M). For instance, the monitoring subgraph G_M presented in Figure 5.6b corresponds to $\alpha_M = 5$, and it has $\lambda_0(G_M) = 18.415$ and $\lambda_1(G_M) = 16.475$, certainly violating the eigenvalue gap criterion in Eq. (5.45), while a randomly generated graph G_{r_M} $(r_M = 17)$, with the same average degree as graph G_M, has $\lambda_0(G_{r_M}) = 17$ and $\lambda_1(G_{r_M}) = 7.633$ satisfying the criterion.

This gap in the graph expansion property does not allow direct application of the scaling law (described by Eq. (5.37) and Eq. (5.38)) to the composite graph (G_M, G_R), which is induced by switching capacity limit α_M and α_R, even though (G_M, G_R) have the the same average degree as its approximation (G_{r_M}, G_{r_R}) by construction. Therefore, we employ simulation to see if the approximation is valid.

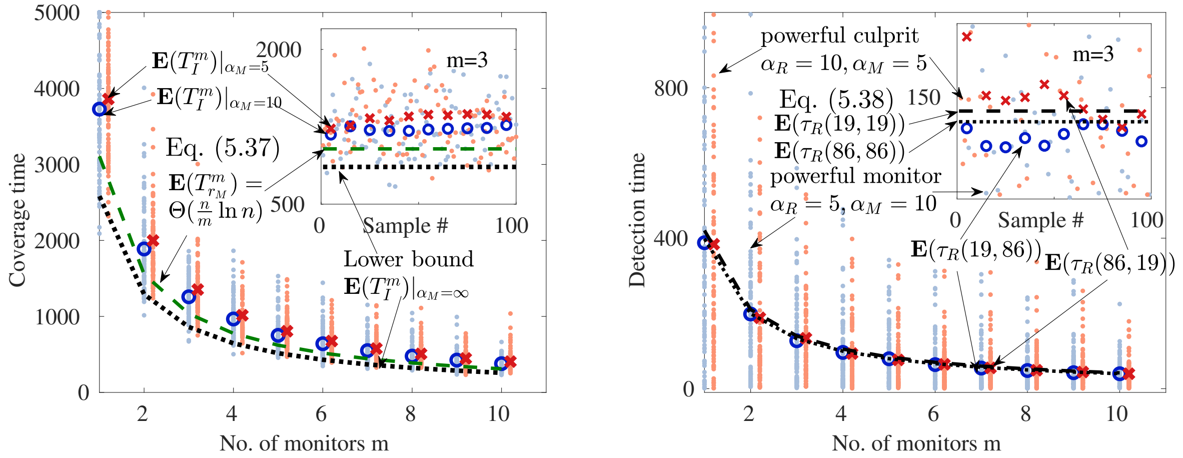
5.6.3 Numerical Results

We validate the regular approximation in the same assignment space V detailed in Table 5.4. Simulation results (dots) and bounds (dashed and dotted lines) of the coverage time and detection time, under an I-strategy with different switching capacities, are shown in Figure 5.14a and 5.14b, respectively. The powerful monitors case ($\alpha_M = 10$, corresponding to $r_M = 19$, and $\alpha_R = 5$, corresponding to $r_R = 86$) is marked in blue, whose mean is shown by blue '○' marker, while the powerful culprit case ($\alpha_M = 5$, $\alpha_R = 10$) is marked in red, whose mean is shown by red '×' marker. In Figure 5.14a, the lower bound of the coverage time (black dotted line) is obtained by setting r_M to ∞ in Eq. (5.37).

From the coverage time in Figure 5.14a, we observe as we anticipated: i) The coverage time of a weaker monitor set (red '×' markers, $r_M = 19$) is slightly longer than that of a more powerful monitor set (blue '○' markers, $\alpha_M = 10$). ii) The $\Theta(\frac{n}{m} \ln n)$ scaling law of the expected coverage time $\mathbb{E}(T_I^m)$ over m is well captured, despite the switching capacity limit.

From the detection time in Figure 5.14b: i) as predicted by Corollary 5.1, the expected detection time $\mathbb{E}(\tau_R(f_I^m))|_{\alpha_M=5}$ is not lengthened much compared to the strengthened case $\alpha_M = 10$, as opposed to an intuitive anticipation, which indicates that both the time and range aspects of the switching capacity do not impact the expected detection time much. ii) Both the upper and lower bound of the expected detection time are tight, if not precise ($\mathbb{E}(\tau_R(f_I^M)) \simeq \mathbb{E}(\tau_R(r_R, r_M))$ for $m \in [1, 10]$).

From both figures: i) Even though the switching capacity of monitors (α_M) and that of the culprit (α_R) differ considerably in value for the two simulation cases, the mean coverage and detection time (round and '×' markers in both Figure 5.14a and Figure 5.14b) are pretty close.



(a) Coverage time $\mathbb{E}(T_I^m) = \Theta(\frac{n}{m}\ln n)$.

(b) Detection time $\mathbb{E}(\tau_a(f_I^m)) = \Theta(\frac{n}{m})$.

Figure 5.14 The expected coverage time and detection time for SAS processes with switching capacity limits (Bounds are derived with regular graph approximation).

The reason behind this is similar to what is revealed in Lemma 5.1, *i.e.*, the more 'mobile' (either monitors or culprits), the more 'visible' to spectrum monitors. ii) Through (G_M, G_R) are not regular graphs, bounds derived for their regular graph approximation (G_{r_M}, G_{r_R}) (dash and dotted black lines) apply smoothly to both the coverage time, and detection time, in the sense that the scaling laws are well-captured in both figures.

Comparing unlimited (Figure 5.11a and 5.12a) with limited (Figure 5.14a and 5.14b) switching capacity cases, the capability limit α_M becomes less influential as the number of monitors m increases, and does not change the scaling behavior over m. The reason behind this is that α_M is sufficiently large so that the quantity $\frac{r_M-1}{r_M-2}$ in Eq. (5.37) comes close to 1. With extensive simulation, we found that $\mathbb{E}(T_I^m)$ and $\mathbb{E}(\tau_R(f_I^m))$ on the real composite graph (G_M, G_R) actually follow the $\Theta(\frac{n\ln n}{m})$ and $\Theta(\frac{n}{m})$ scaling law described in Eq. (5.37) and Eq. (5.38)). We speculate the reason is that both subgraphs (G_M and G_R) are well-connected in degree sense, and the variation in node degree is small, such that G_M and G_R are 'regular' enough. On the other hand, this observation makes us wonder whether the requirement of being 'nice' is necessary in achieving the $\Theta(\frac{n\ln n}{m})$ and $\Theta(\frac{n}{m})$ scaling law.

5.7 Summary

From the impact aspect of mobile data, in this chapter, we examine spectrum activity surveillance (SAS) for DSA-enabled wireless systems, which are envisioned as the monumental delivery network for many data services. We identify the two main objectives of SAS, as sweep-coverage of spectra-location space, and detection of simple/adversarial spectrum culprits, for which we define quantitative metrics, such that any SAS monitor deployment strategy can be fairly evaluated. To accurately describe SAS processes, we introduce a 3-D model that incorporates spectra, temporal and geographical domains, and captures the locality of different spectrum activities. Based on this model, we formulate a SAS process into a graph walk, by space tessellation and pruning with the switching capacity limits of monitors. As an application of the proposed model, we present a deterministic strategy to achieve low switching cost, and randomized strategies to quickly catch adversarial culprits. Efficacy of these strategies are theoretically analyzed and validated through simulations, revealing how the size of a SAS system and the amount of monitoring resource affect their performance. We hope these results contribute to the knowledge of spectrum surveillance, and benefit the design of DSA-enabled wireless systems.

Chapter 6

Conclusion and Future Directions

In this dissertation, we have presented our results on mobile data dynamics in heterogeneous wireless networks, from four aspects: the cause, description, governing rule, and impact. Next, we summarize our main results and discuss the possible future extensions.

6.1 Conclusion

This dissertation discussed the four aspects of mobile data in a top-down manner. We first modeled the propagation of conflicting information, to answer *when* data start and stop moving in networks in Chapter 2. Then in Chapter 3, we studied *where* data can be accessed, during its dissemination process, by modeling data coverage as a graph signal. Onto the governing rule of mobile data, *i.e.*, *how* data move and *what impact* do mobile data cause, we investigated the task offloading process in the resource-constrained fog paradigm, in Chapter 4. Finally, we examined spectrum activity surveillance (SAS), to observe the *impact* of mobile data in Chapter 5. We recapitulate our major findings on mobile data as follows.

In Chapter 2, we studied the driving force of mobile data, that is, information dynamics. Specifically, we proposed a Susceptible-Infected-Cured epidemic model to study the conflicting information propagation phenomenon in networks, particularly the transient competition between the two, during the evolution. Our analysis revealed the impact of network topology, propagation parameters, and the initial conditions, on the lifetime of the undesired information in a network. Upper bounds of the extinction time indicate a dominant impact of network topology, captured by the Cheeger constant of a network, which determines the scaling laws of the lifetime over the network size. As an application of the model and results, we proposed practical and efficient information control measures, that is, injecting desired information into the network, and designed an inference algorithm to obtain the number of information adopters, before the conflicting information propagation process fully unfolds. Our findings quantitatively revealed the impact of network topology on the lifetime time of mobile data, which contributes to the design, management, and restoration of networked systems.

In Chapter 3, we examined data coverage, that is, locations where data are accessible, during the dissemination process in the heterogeneous wireless network. We defined data-strength to quantify data coverage as a numeric signal on a graph, which captures user mobility and geographical adjacency. Then, with graph signal processing tools, we observe high, decreasing, but bounded impact, of user mobility on the change of data coverage. Based on this observation and further analysis, we built a prediction framework, which can estimate future data coverage based on historic observations. Our results contribute to the understanding of data's movements and whereabouts, which are closely related to data service provisioning in such heterogeneous networks, while the graph signal representation of data coverage permits further examination, on the rich temporal and spatial properties of the coverage dynamics.

In Chapter 4, we investigated the task offloading process in the fog paradigm, which is a resource-constrained system residing at the network edge. We studied how multiple tasks, as a special type of data blocks, move during offloading processes given the resource constraint, as well as the impact, *i.e.*, resource demand, of mobile data on the provisioning networks. We proposed a gravity-based task offloading model, which captures the probabilistic offloading decisions based on various criteria, and defined device and network efforts with respect to data movements, to quantify the resource need for the massive task offloading procedures in fog. With this model, we found that the time to complete individual tasks decreases with network size, while the total resource consumption by the system scales linearly with the network size. Our findings address the scalability issue of fog, in terms of resource consumption, which is especially meaningful for large-scale fog applications, such as IoT.

In Chapter 5, we analyzed spectrum activity surveillance (SAS) for DSA-enabled wireless systems, in order to observe the impact of mobile data on radio spectrum, which is a scarce resource at the network edge. We studied this problem in a 3-D space that incorporates spectra, temporal and geographical domains, and formulated the surveillance process into tractable graph walks. We defined two performance metrics, namely the coverage time and detection time, for spectrum monitor deployment strategies, such that any strategy can be fairly evaluated and compared. For the surveillance scenarios by dedicated spectrum monitors, and that by crowd-source spectrum monitors, we designed low-cost deterministic strategy for fast surveillance coverage, and effective randomized strategies for quick detection of spectrum culprits. Our results provide useful design guidelines for large-scale spectrum surveillance systems.

6.2 Future Directions

The work presented in this dissertation is only part of the efforts toward understanding mobile data dynamics in heterogeneous wireless networks. In order to provide efficient, reliable, and accessible data services in future wireless networks, which is expected to host numerous heterogeneous wireless devices, our study may be extended in the following directions.

Due to the open nature of current and future wireless systems, data are not only mobile in the geographical space domain, which is studied in this dissertation, but also mutating in content, representation, and so on. A natural follow-up question is, how does such mutation affect the mobility and accessibility of data? For example, a semantically complete data block can be fragmented and/or encrypted, when it is disseminated through the network. In this case, the data coverage, where this data block is accessible, becomes a process on a higher-dimensional structure that captures location, access capability, and data integrity, instead of the 2-D graph studied in Chapter 3. It would be an interesting research problem to model these aspects in a data dissemination process, for the benefit of data service provisioning.

Considering the exponentially growing size of wireless systems, powered by the proliferating wireless devices, the speed of network expansion (in size) may well surpass the speed of network capacity increase in a wireless system. For instance, in the fog paradigm, our work in Chapter 4 revealed that the growing number of offloading service participants will result in a linearly growing demand on network resources, in the form of radio spectrum, resource blocks, and so on. However, the total amount of network resource in the current wireless system is upper bounded, which indicates that the scale of data offloading service can not grow indefinitely. Consequently, it is highly desirable to obtain such limit, which is an important characteristic of a wireless system, and a key input for designing data services. On the other hand, spectrum dynamics, as a result of mobile data, in turn limits the mobility of data, especially in densely populated networks, such as a fog system designed for IoT applications. Consequently, it will be equally necessary to study the inverse of the impact problem addressed in Chapter 5: what is the impact of spectrum dynamics on mobile data?

REFERENCES

[1] Rohde & schwarz® fsh4/8/13/20 spectrum analyzer operation manual. https://www.rohde-schwarz.com/us/manual/r-s-fsh4-8-13-20-operating-manual-manuals-gb1_78701-29159.html, 2016. Accessed: 2018-2-21.

[2] Ieee standard for adoption of openfog reference architecture for fog computing. *IEEE Std 1934-2018*, pages 1–176, Aug 2018.

[3] 3GPP. NR; User Equipment (UE) radio transmission and reception; Part 3: Range 1 and Range 2 Interworking operation with other radios. Technical Specification (TS) 38.101-3, 3rd Generation Partnership Project (3GPP), 01 2019. Version 15.4.0.

[4] Yuan Ai, Mugen Peng, and Kecheng Zhang. Edge computing technologies for internet of things: a primer. *Digital Communications and Networks*, 4(2):77 – 86, 2018.

[5] David Aldous and James Allen Fill. Reversible markov chains and random walks on graphs, 2002. Unfinished monograph, recompiled 2014, available at https://www.stat.berkeley.edu/~aldous/RWG/book.pdf.

[6] Harriet Alexander. Boston bomb: Reddit apologises for identifying wrong suspects. http://www.telegraph.co.uk/news/worldnews/northamerica/usa/10012382/, 2013.

[7] Noga Alon, Chen Avin, Michal Koucky, Gady Kozma, Zvi Lotker, and Mark R. Tuttle. Many random walks are faster than one. *Combinatorics, Probability and Computing*, 20(4):481–502, 2011.

[8] G. Andrienko, N. Andrienko, W. Chen, R. Maciejewski, and Y. Zhao. Visual analytics of mobility and transportation: State of the art and further research directions. *IEEE Transactions on Intelligent Transportation Systems*, 18(8):2232–2249, Aug 2017.

[9] Ramon Aparicio-Pardo, Karine Pires, Alberto Blanc, and Gwendal Simon. Transcoding live adaptive video streams at a massive scale in the cloud. In *Proceedings of the 6th ACM Multimedia Systems Conference*, MMSys '15, pages 49–60, New York, NY, USA, 2015. ACM.

[10] Terje Aven. Upper (lower) bounds on the mean of the maximum (minimum) of a number of random variables. *Journal of Applied Probability*, 22(3):723–728, 1985.

[11] Fan Bai and Ahmed Helmy. A survey of mobility models. *Wireless Adhoc Networks. University of Southern California, USA*, 206:147, 2004.

[12] G. Baldini and G. Steri. A survey of techniques for the identification of mobile phones using the physical fingerprints of the built-in components. *IEEE Communications Surveys Tutorials*, 19(3):1761–1789, thirdquarter 2017.

[13] C. Baylis, M. Fellows, L. Cohen, and R. J. Marks II. Solving the spectrum crisis: Intelligent, reconfigurable microwave transmitter amplifiers for cognitive radar. *IEEE Microwave Magazine*, 15(5):94–107, July 2014.

[14] Tolga Bektas. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3), 2006.

[15] Dimitris Bertsimas, Karthik Natarajan, and Chung-Piaw Teo. Tight bounds on expected order statistics. *Probab. Eng. Inf. Sci.*, 20(4):667–686, October 2006.

[16] Alex Beutel, B. Aditya Prakash, Roni Rosenfeld, and Christos Faloutsos. Interacting viruses in networks: Can both survive? In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 426–434. ACM, 2012.

[17] D. Bienstock. Optimal control of cascading power grid failures. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 2166–2173, Dec 2011.

[18] Kashif Bilal, Osman Khalid, Aiman Erbad, and Samee U. Khan. Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers. *Computer Networks*, 130:94 – 120, 2018.

[19] Christian Borgs, Jennifer Chayes, Ayalvadi Ganesh, and Amin Saberi. How to distribute antidote to control epidemics. *Random Structures & Algorithms*, 37(2):204–222, 2010.

[20] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163–177, 2001.

[21] L. Caccetta and W.F. Smyth. Graphs of maximum diameter. *Discrete Mathematics*, 102(2):121 – 141, 1992.

[22] Deepayan Chakrabarti, Yang Wang, Chenxi Wang, Jurij Leskovec, and Christos Faloutsos. Epidemic thresholds in real networks. *ACM Trans. Inf. Syst. Secur.*, 10(4):1:1–1:26, January 2008.

[23] Lixing Chen and Jie Xu. Socially trusted collaborative edge computing in ultra dense networks. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, SEC '17, pages 9:1–9:11, New York, NY, USA, 2017. ACM.

[24] Min Chen and Yixue Hao. Task offloading for mobile edge computing in software defined ultra-dense network. *IEEE Journal on Selected Areas in Communications*, 36(3):587–597, 2018.

[25] P. Y. Chen and K. C. Chen. Optimal control of epidemic information dissemination in mobile ad hoc networks. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–5, Dec 2011.

[26] Y. Chen, M. Ding, D. Lopez-Perez, J. Li, Z. Lin, and B. Vucetic. Dynamic reuse of unlicensed spectrum: An inter-working of lte and wifi. *IEEE Wireless Communications*, 24(5):52–59, October 2017.

[27] Zesheng Chen and Chuanyi Ji. Spatial-temporal modeling of malware propagation in networks. *Neural Networks, IEEE Transactions on*, 16(5):1291–1303, Sept 2005.

[28] F. Chiti, R. Fantacci, and B. Picano. A matching theory framework for tasks offloading in fog computing for iot systems. *IEEE Internet of Things Journal*, 5(6):5089–5096, Dec 2018.

[29] Fan RK Chung. Laplacians of graphs and cheeger's inequalities. *Combinatorics, Paul Erdos is Eighty*, 2(157-172):13–2, 1996.

[30] Cisco. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022. Technical report, 02 2019.

[31] Colin Cooper and Alan Frieze. The cover time of random regular graphs. *SIAM Journal on Discrete Mathematics*, 18(4):728–740, 2005.

[32] Colin Cooper, Alan Frieze, and Tomasz Radzik. *Multiple Random Walks and Interacting Particle Systems*, pages 399–410. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[33] A. Dadlani, M. S. Kumar, M. G. Maddi, and K. Kim. Mean-field dynamics of inter-switching memes competing over multiplex social networks. *IEEE Communications Letters*, 21(5):967–970, May 2017.

[34] T. D. Dang and D. Hoang. Data mobility as a service. In *2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 67–71, June 2016.

[35] Gautam Dasarathy. A simple probability trick for bounding the expected maximum of n random variables. `https://www.ece.rice.edu/~gd14/files/maxGaussians.pdf`, 2011.

[36] Marie-Catherine De Marneffe, Anna N Rafferty, and Christopher D Manning. Finding contradictions in text. In *ACL*, volume 8, pages 1039–1047, 2008.

[37] B. Van den Bergh, D. Giustiniano, H. Cordobés, M. Fuchs, R. Calvo-Palomino, S. Pollin, S. Rajendran, and V. Lenders. Electrosense: Crowdsourcing spectrum monitoring. In *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 1–2, March 2017.

[38] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, and V. K. Bhargava. A survey on non-orthogonal multiple access for 5g networks: Research challenges and future trends. *IEEE Journal on Selected Areas in Communications*, 35(10):2181–2195, Oct 2017.

[39] Daniel B Faria. Modeling signal attenuation in ieee 802.11 wireless lans. Technical report, Stanford Univeristy, 2005.

[40] Mylynn Felt. Social media and the social sciences: How researchers employ big data analytics. *Big Data & Society*, 3(1):2053951716645828, 2016.

[41] Ian Fogg. The State of Wifi vs Mobile Network Experience as 5G Arrives. Technical report, OpenSignal, 11 2018.

[42] A. Ganesh, L. Massoulie, and D. Towsley. The effect of network topology on the spread of epidemics. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 2, pages 1455–1466 vol. 2, March 2005.

[43] Maria Garnaeva, Victor Chebyshev, Denis Makrushin, Roman Unuchek, and Anton Ivanov. Kaspersky security bulletin 2014. overall statistics for 2014. http://securelist.com/analysis/kaspersky-security-bulletin/68010/kaspersky-security-bulletin-2014-overall-statistics-for-2014/, 2014.

[44] Google. Google spectrum database: Google earth visualization of available tv white space spectrum. `https://www.google.com/get/spectrumdatabase/`, 2013. Accessed: 2017-12-18.

[45] Francesco Grassi, Andreas Loukas, Nathanael Perraudin, and Benjamin Ricaud. A time-vertex signal processing framework: Scalable processing and meaningful representations for time-series on graphs. *Trans. Sig. Proc.*, 66(3):817–829, February 2018.

[46] M. Grossglauser and D. N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking*, 10(4):477–486, Aug 2002.

[47] Arman Hasanzadeh, Xi Liu, Nick Duffield, Krishna R. Narayanan, and Byron Chigoy. A graph signal processing approach for real-time traffic prediction in transportation networks. *ICASSP 2018 (submitted)*, /abs/1711.06954, 2017.

[48] Dale N. Hatfield, Lynn Claudy, Mark Gorenberg, Dave Gurney, Greg Lapin, Brian Markwalter, Geoffrey Mendenhall, Pierre de Vries, and Dennis Roberson. Introduction to

interference resolution, enforcement and radio noise. Technical Report FCC 14-31, FCC Technological Advisory Council, Jun. 2014.

[49] J. He, J. Wei, K. Chen, Z. Tang, Y. Zhou, and Y. Zhang. Multitier fog computing with large-scale iot data analytics for smart cities. *IEEE Internet of Things Journal*, 5(2):677–686, April 2018.

[50] Oliver Holland, Hanna Bogucka, and Arturas Medeisis. *Practical Mechanisms Supporting Spectrum Sharing*, pages 450–. Wiley Telecom, 2015.

[51] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[52] Y. Huo, X. Dong, and W. Xu. 5g cellular user equipment: From theory to practical hardware design. *IEEE Access*, 5:13992–14010, 2017.

[53] M. Höyhtyä, A. Mämmelä, M. Eskola, M. Matinmikko, J. Kalliovaara, J. Ojaniemi, J. Suutala, R. Ekman, R. Bacchus, and D. Roberson. Spectrum occupancy measurements: A survey and use of interference maps. *IEEE Communications Surveys Tutorials*, 18(4):2386–2414, Fourthquarter 2016.

[54] Mayank Jain, Jung Il Choi, Taemin Kim, Dinesh Bharadia, Siddharth Seth, Kannan Srinivasan, Philip Levis, Sachin Katti, and Prasun Sinha. Practical, real-time, full duplex wireless. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, MobiCom '11, pages 301–312, New York, NY, USA, 2011. ACM.

[55] Y. Jiang, Y. Chen, S. Yang, and C. Wu. Energy-efficient task offloading for time-sensitive applications in fog computing. *IEEE Systems Journal*, 13(3):2930–2941, Sep. 2019.

[56] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang. Specguard: Spectrum misuse detection in dynamic spectrum access systems. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 172–180, April 2015.

[57] M.H.R. Khouzani, S. Sarkar, and E. Altman. Optimal control of epidemic evolution. In *INFOCOM, 2011 Proceedings IEEE*, pages 1683–1691, April 2011.

[58] Mijung Kim and K. Selçuk Candan. Sbv-cut: Vertex-cut based graph partitioning using structural balance vertices. *Data & Knowledge Engineering*, 72:285 – 303, 2012.

[59] Subhashini Krishnasamy, Siddhartha Banerjee, and Sanjay Shakkottai. The behavior of epidemics under bounded susceptibility. In *The 2014 ACM International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '14, pages 263–275, New York, NY, USA, 2014. ACM.

[60] Dave Lee. Boston bombing: How internet detectives got it very wrong. `http://www.bbc.com/news/technology-22214511`, 2013.

[61] Marc Lelarge. Efficient control of epidemics over random networks. In *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '09, pages 1–12, New York, NY, USA, 2009. ACM.

[62] Adam Lella and Andrew Lipsman. The 2016 u.s. cross-platform future in focus (comscore whitepaper). `https://www.comscore.com/Insights/Presentations-and-Whitepapers/2016/2016-US-Cross-Platform-Future-in-Focus`, March 2016.

[63] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 497–506, New York, NY, USA, 2009. ACM.

[64] Jure Leskovec and Julian J. Mcauley. Learning to discover social circles in ego networks. In P. Bartlett, F.c.n. Pereira, C.j.c. Burges, L. Bottou, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 548–556. 2012.

[65] M. Li, D. Yang, J. Lin, M. Li, and J. Tang. Specwatch: Adversarial spectrum usage monitoring in crns with unknown statistics. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, April 2016.

[66] Y. Li and W. Wang. Horizon on the move: Geocast in intermittently connected vehicular ad hoc networks. In *2013 Proceedings IEEE INFOCOM*, pages 2553–2561, April 2013.

[67] Y. Li and W. Wang. Can mobile cloudlets support mobile applications? In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 1060–1068, April 2014.

[68] Z. Li, C. Wang, S. Yang, C. Jiang, and I. Stojmenovic. Space-crossing: Community-based data forwarding in mobile social networks under the hybrid communication architecture. *IEEE Transactions on Wireless Communications*, 14(9):4720–4727, Sept 2015.

[69] C. Liang and F. R. Yu. Wireless network virtualization: A survey, some research issues and challenges. *IEEE Communications Surveys Tutorials*, 17(1):358–380, Firstquarter 2015.

[70] Yishi Lin, J.C.S. Lui, Kyomin Jung, and Sungsu Lim. Modeling multi-state diffusion process in complex networks: Theory and applications. In *Signal-Image Technology Internet-Based Systems (SITIS), 2013 International Conference on*, pages 506–513, Dec 2013.

[71] Song Liu, Larry J. Greenstein, Wade Trappe, and Yingying Chen. Detecting anomalous spectrum usage in dynamic spectrum access networks. *Ad Hoc Networks*, 10(5):831 – 844, 2012. Special Issue on Cognitive Radio Ad Hoc Networks.

[72] Madhusanka Liyanage, Andrei Gurtov, and Mika Ylianttila. *Leveraging SDN for the 5G Networks*, pages 440–. Wiley Telecom, 2015.

[73] Andrea Lottarini, Alex Ramirez, Joel Coburn, Martha A Kim, Parthasarathy Ranganathan, Daniel Stodolsky, and Mark Wachsler. vbench: Benchmarking video transcoding in the cloud. In *ACM SIGPLAN Notices*, volume 53, pages 797–809. ACM, 2018.

[74] Anand Louis. *The complexity of expansion problems*. PhD thesis, Georgia Institute of Technology, 2014.

[75] A. Loukas, E. Isufi, and N. Perraudin. Predicting the evolution of stationary graph signals. In *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pages 60–64, Oct 2017.

[76] Z. Lu, W. Wang, and C. Wang. How can botnets cause storms? understanding the evolution and impact of mobile botnets. In *INFOCOM, 2014 Proceedings IEEE*, pages 1501–1509, April 2014.

[77] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PLOS ONE*, 13(3):1–26, 03 2018.

[78] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys Tutorials*, 19(4):2322–2358, Fourthquarter 2017.

[79] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro. Stationary graph processes and spectral estimation. *IEEE Transactions on Signal Processing*, 65(22):5911–5926, Nov 2017.

[80] Microsoft. White spaces database. `http://whitespaces.microsoftspectrum.com/`, 2015. Accessed: 2018-02-11.

[81] M Miller and Jozef Siran. Moore graphs and beyond: A survey of the degree/diameter problem. *Electronic Journal of Combinatorics (Dynamic Surveys)*, pages 1–92, May 2013.

[82] Jonny Milliken, Valerio Selis, and Alan Marshall. Detection and analysis of the chameleon wifi access point virus. *EURASIP Journal on Information Security*, 2013(1):2, 2013.

[83] M. E. J. Newman. Threshold Effects for Two Pathogens Spreading on a Network. *Physical Review Letters*, 95(10):108701, September 2005.

[84] Ana Nika, Zengbin Zhang, Xia Zhou, Ben Y. Zhao, and Haitao Zheng. Towards commoditized real-time spectrum monitoring. In *Proceedings of the 1st ACM Workshop on Hot Topics in Wireless*, HotWireless '14, pages 25–30, New York, NY, USA, 2014. ACM.

[85] Huansheng Ning. *Unit and Ubiquitous Internet of Things*. CRC Press, 2013.

[86] C. Nowzari, V. M. Preciado, and G. J. Pappas. Analysis and control of epidemics: A survey of spreading processes on complex networks. *IEEE Control Systems Magazine*, 36(1):26–46, Feb 2016.

[87] M. J. L. Pan, T. C. Clancy, and R. W. McGwier. A machine learning approach for dynamic spectrum access radio identification. In *2014 IEEE Global Communications Conference*, pages 1041–1046, Dec 2014.

[88] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani. Epidemic processes in complex networks. *Reviews of Modern Physics*, 87:925–979, July 2015.

[89] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic spreading in scale-free networks. *Physical review letters*, 86(14):3200, 2001.

[90] C. Perera, C. H. Liu, and S. Jayawardena. The emerging internet of things marketplace from an industrial perspective: A survey. *IEEE Transactions on Emerging Topics in Computing*, 3(4):585–598, Dec 2015.

[91] Damian Pfammatter, Domenico Giustiniano, and Vincent Lenders. A software-defined sensor architecture for large-scale wideband spectrum monitoring. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, IPSN '15, pages 71–82, New York, NY, USA, 2015. ACM.

[92] Johan Philip. *The probability distribution of the distance between two random points in a box*. KTH mathematics, Royal Institute of Technology, 2007.

[93] Michal Piorkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. CRAWDAD dataset epfl/mobility (v. 2009-02-24). Downloaded from `http://crawdad.org/epfl/mobility/20090224`, February 2009.

[94] B. Aditya Prakash, Alex Beutel, Roni Rosenfeld, and Christos Faloutsos. Winner takes all: Competing viruses or ideas on fair-play networks. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 1037–1046, New York, NY, USA, 2012. ACM.

[95] B.Aditya Prakash, Deepayan Chakrabarti, NicholasC. Valler, Michalis Faloutsos, and Christos Faloutsos. Threshold conditions for arbitrary cascade models on arbitrary networks. *Knowledge and Information Systems*, 33(3):549–575, 2012.

[96] V.M. Preciado, M. Zargham, and D. Sun. A convex framework to control spreading processes in directed networks. In *Information Sciences and Systems (CISS), 2014 48th Annual Conference on*, pages 1–6, March 2014.

[97] L. Pu, X. Chen, J. Xu, and X. Fu. D2d fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted d2d collaboration. *IEEE Journal on Selected Areas in Communications*, 34(12):3887–3901, Dec 2016.

[98] Feng Qi and Bai-Ni Guo. Sharp bounds for harmonic numbers. *arXiv preprint arXiv:1002.3856*, 2010.

[99] J. Qin, H. Zhu, Y. Zhu, L. Lu, G. Xue, and M. Li. Post: Exploiting dynamic sociality for mobile advertising in vehicular networks. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 1761–1769, April 2014.

[100] J. Qin, H. Zhu, Y. Zhu, L. Lu, G. Xue, and M. Li. Post: Exploiting dynamic sociality for mobile advertising in vehicular networks. *IEEE Transactions on Parallel and Distributed Systems*, 27(6):1770–1782, June 2016.

[101] M. A. Rahman, M. S. Hossain, E. Hassanain, and G. Muhammad. Semantic multimedia fog computing and iot environment: Sustainability perspective. *IEEE Communications Magazine*, 56(5):80–87, May 2018.

[102] X. Ren, P. London, J. Ziani, and A. Wierman. Datum: Managing data purchasing and data placement in a geo-distributed data market. *IEEE/ACM Transactions on Networking*, 26(2):893–905, April 2018.

[103] Grand View Research. Global Positioning Systems (GPS) Market Size, Share & Trends Analysis Report By Deployment, By Application (Aviation, Marine, Surveying, Location-Based Services, Road), And Segment Forecasts, 2018 - 2025. Technical report, Grand View Research, 10 2018.

[104] S. Safavi, U. A. Khan, S. Kar, and J. M. F. Moura. Distributed localization: A linear theory. *Proceedings of the IEEE*, 106(7):1204–1223, July 2018.

[105] A. Sandryhaila and J. M. F. Moura. Discrete signal processing on graphs: Frequency analysis. *IEEE Transactions on Signal Processing*, 62(12):3042–3054, June 2014.

[106] Michael Seufert, Tobias Griepentrog, Valentin Burger, and Tobias Hoβsfeld. A simple wifi hotspot model for cities. *IEEE Communications Letters*, 20:384–387, 2016.

[107] Dong-Hoon Shin and Saurabh Bagchi. Optimal monitoring in multi-channel multi-radio wireless mesh networks. In *Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '09, pages 229–238, New York, NY, USA, 2009. ACM.

[108] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

[109] Sppedtest. Speedtest Market Reports, United States of America. Technical report, Ookla, 7 2018.

[110] Charles Steinfield. *The development of location based services in mobile commerce*, pages 177–197. Physica-Verlag HD, Heidelberg, 2004.

[111] Kaarthik Sundar and Sivakumar Rathinam. Generalized multiple depot traveling salesmen problem—polyhedral study and exact algorithm. *Computers & Operations Research*, 70:39 – 55, 2016.

[112] Frank W. Takes and Walter A. Kosters. Computing the eccentricity distribution of large graphs. *Algorithms*, 6(1):100–118, 2013.

[113] L. Tang and S. He. Multi-user computation offloading in mobile edge computing: A behavioral perspective. *IEEE Network*, 32(1):48–53, Jan 2018.

[114] Fei Tao, Qinglin Qi, Ang Liu, and Andrew Kusiak. Data-driven smart manufacturing. *Journal of Manufacturing Systems*, 48:157 – 169, 2018. Special Issue on Smart Manufacturing.

[115] M. N. Tehrani, M. Uysal, and H. Yanikomeroglu. Device-to-device communication in 5g cellular networks: challenges, solutions, and future directions. *IEEE Communications Magazine*, 52(5):86–92, May 2014.

[116] A. Thapaliya and S. Sengupta. Understanding the feasibility of machine learning algorithms in a game theoretic environment for dynamic spectrum access. In *2017 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, pages 1–8, July 2017.

[117] Sir William Thomson. Lxiii. on the division of space with minimum partitional area. *Philosophical Magazine Series 5*, 24(151), 1887.

[118] L. Tong, Y. Li, and W. Gao. A hierarchical edge cloud architecture for mobile computing. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, April 2016.

[119] Luis M. Vaquero and Luis Rodero-Merino. Finding your way in the fog: Towards a comprehensive definition of fog computing. *SIGCOMM Comput. Commun. Rev.*, 44(5):27–32, October 2014.

[120] Cisco VNI. Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper. Technical report, Cisco and/or its affliates, 02 2019.

[121] A. M. Voicu, L. Simić, and M. Petrova. Inter-technology coexistence in a spectrum commons: A case study of wi-fi and lte in the 5-ghz unlicensed band. *IEEE Journal on Selected Areas in Communications*, 34(11):3062–3077, Nov 2016.

[122] Jian-Wei Wang and Li-Li Rong. Cascade-based attack vulnerability on the us power grid. *Safety Science*, 47(10):1332–1336, 2009.

[123] T. Wang, J. Zhou, A. Liu, M. Z. A. Bhuiyan, G. Wang, and W. Jia. Fog-based computing and storage offloading for data synchronization in iot. *IEEE Internet of Things Journal*, pages 1–1, 2019.

[124] X. Wang, W. Wu, and D. Qi. Mobility-aware participant recruitment for vehicle-based mobile crowdsensing. *IEEE Transactions on Vehicular Technology*, 67(5):4415–4426, May 2018.

[125] D. Weaire and R. Phelan. A counter-example to kelvin's conjecture on minimal surfaces. *Philosophical Magazine Letters*, 69(2):107–110, 1994.

[126] Michel Wedel and P.K. Kannan. Marketing analytics for data-rich environments. *Journal of Marketing*, 80(6):97–121, 2016.

[127] Janet Wiener and Nathan Bronson. Facebook's top open data problems. `https://research.fb.com/facebook-s-top-open-data-problems/`, October 2014.

[128] H. Wu. Performance modeling of delayed offloading in mobile wireless environments with failures. *IEEE Communications Letters*, 22(11):2334–2337, Nov 2018.

[129] Y. Xiao and M. Krunz. Qoe and power efficiency tradeoff for fog computing networks with fog node cooperation. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9, May 2017.

[130] T. Yan, W. Zhang, and G. Wang. Dove: Data dissemination to a desired number of receivers in vanet. *IEEE Transactions on Vehicular Technology*, 63(4):1903–1916, May 2014.

[131] Jaewon Yang and Jure Leskovec. Patterns of temporal variation in online media. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 177–186, New York, NY, USA, 2011. ACM.

[132] Lei Yang, Zengbin Zhang, Ben Y. Zhao, Christopher Kruegel, and Haitao Zheng. Enforcing dynamic spectrum access with spectrum permits. In *Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '12, pages 195–204, New York, NY, USA, 2012. ACM.

[133] Y. Yang, X. Chang, Z. Han, and L. Li. Delay-aware secure computation offloading mechanism in a fog-cloud framework. In *2018 IEEE Intl Conf on Parallel Distributed Processing with Applications, Ubiquitous Computing Communications, Big Data Cloud Computing, Social Computing Networking, Sustainable Computing Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, pages 346–353, Dec 2018.

[134] Y. Yang, K. Wang, G. Zhang, X. Chen, X. Luo, and M. Zhou. Meets: Maximal energy efficient task scheduling in homogeneous fog networks. *IEEE Internet of Things Journal*, 5(5):4076–4087, Oct 2018.

[135] J. Yao and N. Ansari. Reliability-aware fog resource provisioning for deadline-driven iot services. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2018.

[136] Shanhe Yi, Cheng Li, and Qun Li. A survey of fog computing: Concepts, applications and issues. In *Proceedings of the 2015 Workshop on Mobile Big Data*, Mobidata '15, pages 37–42, New York, NY, USA, 2015. ACM.

[137] S. Yoon, L. E. Li, S. C. Liew, R. R. Choudhury, I. Rhee, and K. Tan. Quicksense: Fast and energy-efficient channel sensing for dynamic spectrum access networks. In *2013 Proceedings IEEE INFOCOM*, pages 2247–2255, April 2013.

[138] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue. On reducing iot service delay via fog offloading. *IEEE Internet of Things Journal*, 5(2):998–1010, April 2018.

[139] Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, and Jason P. Jue. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of Systems Architecture*, 2019.

[140] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han. Computing resource allocation in three-tier iot fog networks: A joint optimization approach combining stackelberg game and matching. *IEEE Internet of Things Journal*, 4(5):1204–1215, Oct 2017.

[141] H. Zhang, Z. Zhang, and H. Dai. Gossip-based information spreading in mobile networks. *IEEE Transactions on Wireless Communications*, 12(11):5918–5928, November 2013.

[142] K. Zhang, S. Leng, X. Peng, L. Pan, S. Maharjan, and Y. Zhang. Artificial intelligence inspired transmission scheduling in cognitive vehicular communications and networks. *IEEE Internet of Things Journal*, pages 1–1, 2019.