Protocol Design and Performance Evaluation for Wireless Ad Hoc Networks

by

Fei Tong
B.Eng., South-Central University for Nationalities, China, 2009
M.Eng., Chonbuk National University, South Korea, 2011

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Computer Science

© Fei Tong, 2016
University of Victoria

All rights reserved. This dissertation may not be reproduced in whole or in part, by photocopying or other means, without the permission of the author.

Protocol Design and Performance Evaluation
for Wireless Ad Hoc Networks

by

Fei Tong
B.Eng., South-Central University for Nationalities, China, 2009
M.Eng., Chonbuk National University, South Korea, 2011

Supervisory Committee

Dr. Jianping Pan, Supervisor
(Department of Computer Science)

Dr. Kui Wu, Departmental Member
(Department of Computer Science)

Dr. Yang Shi, Outside Member
(Department of Mechanical Engineering)

Supervisory Committee

Dr. Jianping Pan, Supervisor
(Department of Computer Science)

Dr. Kui Wu, Departmental Member
(Department of Computer Science)

Dr. Yang Shi, Outside Member
(Department of Mechanical Engineering)

Abstract

Benefiting from the constant and significant advancement of wireless communication technologies and networking protocols, Wireless Ad hoc NETwork (WANET) has played a more and more important role in modern communication networks without relying much on existing infrastructures. The past decades have seen numerous applications adopting ad hoc networks for service provisioning. For example, Wireless Sensor Network (WSN) can be widely deployed for environment monitoring and object tracking by utilizing low-cost, low-power and multi-function sensor nodes. To realize such applications, the design and evaluation of communication protocols are of significant importance. Meanwhile, the network performance analysis based on mathematical models is also in great need of development and improvement.

This dissertation investigates the above topics from three important and fundamental aspects, including data collection protocol design, protocol modeling and analysis, and physical interference modeling and analysis. The contributions of this dissertation are four-fold.

First, this dissertation investigates the synchronization issue in the duty-cycled, pipelined-scheduling data collection of a WSN, based on which a pipelined data collection protocol, called PDC, is proposed. PDC takes into account both the pipelined data collection and the underlying schedule synchronization over duty-cycled radios

practically and comprehensively. It integrates all its components in a natural and seamless way to simplify the protocol implementation and to achieve a high energy efficiency and low packet delivery latency. Based on PDC, an Adaptive Data Collection (ADC) protocol is further proposed to achieve dynamic duty-cycling and free addressing, which can improve network heterogeneity, load adaptivity, and energy efficiency. Both PDC and ADC have been implemented in a pioneer open-source operating system for the Internet of Things, and evaluated through a testbed built based on two hardware platforms, as well as through emulations.

Second, Linear Sensor Network (LSN) has attracted increasing attention due to the vast requirements on the monitoring and surveillance of a structure or area with a linear topology. Being aware that, for LSN, there is few work on the network modeling and analysis based on a duty-cycled MAC protocol, this dissertation proposes a framework for modeling and analyzing a class of duty-cycled, multi-hop data collection protocols for LSNs. With the model, the dissertation thoroughly investigates the PDC performance in an LSN, considering both saturated and unsaturated scenarios, with and without retransmission. Extensive OPNET simulations have been carried out to validate the accuracy of the model.

Third, in the design and modeling of PDC above, the transmission and interference ranges are defined for successful communications between a pair of nodes. It does not consider the cumulative interference from the transmitters which are out of the contention range of a receiver. Since most performance metrics in wireless networks, such as outage probability, link capacity, etc., are nonlinear functions of the distances among communicating, relaying, and interfering nodes, a physical interference model based on distance is definitely needed in quantifying these metrics. Such quantifications eventually involve the Nodal Distance Distribution (NDD) intrinsically depending on network coverage and nodal spatial distribution. By extending a tool in integral geometry and using decomposition and recursion, this dissertation proposes a systematic and algorithmic approach to obtaining the NDD between two nodes which are uniformly distributed at random in an arbitrarily-shaped network.

Fourth, with the proposed approach to NDDs, the dissertation presents a physical interference model framework to analyze the cumulative interference and link outage probability for an LSN running the PDC protocol. The framework is further applied to analyze 2D networks, i.e., ad hoc Device-to-Device (D2D) communications underlaying cellular networks, where the cumulative interference and link outage probabilities for both cellular and D2D communications are thoroughly investigated.

Contents

List of Tables

List of Figures

List of Abbreviations

ADC	Adaptive Data Collection
BPP	Binomial Point Process
BS	Base Station
CCA	Clear Channel Assessment
CCP	Contiki Collect Protocol
CDF	Cumulative Distribution Function
CLD	Chord Length Distribution
CT	Child Table
CTP	Collection Tree Protocol
CUE	Cellular User Equipment
D&R	Decomposition and Recursion
D2D	Device-to-Device
DDC	Dynamic Duty-Cycling
DUE	D2D User Equipment
DPSync	Duty-cycled, Pipelined Synchronization
FA	Free Addressing
GDPS	Grade Division and Pipelined Scheduling
GR	Guard Region
ILA	Interference-Limited Area
KM	Kinematic Measure
k–NN	kth Nearest Neighbor
LSN	Linear Sensor Network
NDD	Nodal Distance Distribution
OS	Operating System
PCTS	PDC's CTS
PDC	Pipelined Data Collection
PDF	Probability Distribution Function

PGI	Packet Generation Interval
PPP	Poisson Point Process
PRTS	PDC's RTS
PT	Parent Table
Ran2Ran NDD	NDD from a random point to another random point
RDC	Radio Duty Cycling
Ref2Ran NDD	NDD from a reference point to a random point
RID	Randomly-generated IDentifier
RV	Random Variable
SIC	Successive Interference Cancellation
SINR	Signal-to-Interference-plus-Noise Ratio
SSL	Sleeping SLots
UE	User Equipment
WANET	Wireless Ad Hoc NETwork
WLOG	Without Loss Of Generality
WSN	Wireless Sensor Network

Acknowledgments

It would not have been possible for me to carry out the work reflected in this dissertation without the support, friendship, and stimulation of many people. I first want to express that I draw much strength from the support of my family and from the tenets of the excellent culture of my motherland. I am fortunate to have you as so much a part of my life. I would not have dared to undertake the exploration of the future without knowing that I could always rely on your endless love and support.

To my dear supervisor, Dr. Jianping Pan, go my sincerest thanks and deepest appreciation. Thank him for always providing me with his great help, support, patience, and encouragement on both my research and life during the past four years. Also I greatly thank Dr. Wan Tang from South-Central University for Nationalities (Wuhan, Hubei, China) who has been my mentor and friend since my undergraduate study, for continually offering her great help, support, and encouragement to me. I would also like to express my appreciation to Dr. Lin Cai from the ECE Department at University of Victoria (UVic), for her constructive instruction and guidance.

Thank Dr. Kui Wu and Dr. Yang Shi, who are my supervisory committee members, and Dr. Hossam Saad Hassanein, who is my external examiner, for their time and expertise to better my work. Especially thank Dr. Kui Wu for his support on the testbed implementation in my work.

Many thanks also go to Dr. Ruonan Zhang from Northwestern Polytechnical University, for hosting me and providing me with his great help and support during my visiting to his lab for research collaborations. Thank Dr. Jiagao Wu and Dr. Linfeng Liu from Nanjing Post and Communications University, and Dr. Jun Tao and Dr. Ming Ling from Southeast University for either hosting me when I visited them or sharing their valuable research and life experience with me. Thank the other professors and friends who I do not mention here for hosting me when I visited them.

Thank my fellow lab-mates and friends in or out of UVic. Your friendships and supports have made my UVic life educational, enjoyable, and memorable.

Thank my landlady and landlord, the couple Mrs. Hua Bai and Mr. Guiping Liu, for providing me with a very peaceful and comfortable accommodation like home.

My special thanks finally go to the couple Mrs. Xiang Zhu and Mr. Songtao Zhang, and Ms. Xiaoying Long.♡.☺.

Fei Tong, Victoria, BC, Canada

Dedication

To My Dear Family, Mentors, Friends, and Motherland.

Chapter 1

Introduction

1.1 Background

Wireless Ad hoc NETwork (WANET), in contrast to the centralized hub-and-spoke networks which heavily rely on pre-existing infrastructures (such as routers/switches in wired networks or base stations/access points in wireless networks), allows low-cost, seamless integration into existing networks with fast deployment. Such networks can provide users with flexible and ubiquitous network access to various services. As one of the most significant applications of wireless communication technologies, a WANET consists of autonomous or mobile nodes which communicate with each other without a centralized control or assistance. All the nodes in the network can transmit, receive and forward messages, and thus do not rely on backbone networks. Therefore, ad hoc networks provide more robustness and flexibility in the presence of node failures than those requiring infrastructure support and are quite useful in environment monitoring, infrastructure surveillance, disaster relief, battlefield, and scientific exploration.

Recent advances in wireless sensor networking technologies now facilitate various services and allow the implementation of low-cost, pervasive, flexible and rapidly-deployed monitoring and control systems. A typical Wireless Sensor Network (WSN) usually consists of a large number of sensor nodes randomly deployed with one or a few sinks as the gateway(s) of the network. The unattended sensor nodes are tightly constrained in terms of energy, processing, and storage capacities. Especially, efficient energy utilization is of crucial importance in WSNs, due to the limited battery capacity of sensor nodes.

Data collection can be found in a wide range of applications of WSNs, such as scientific exploration [1], and other event monitoring applications [2–4]. Sensor nodes send data packets usually through multi-hop wireless communications to a common sink node for further processing, characterized with the many-to-one traffic pattern. The past decades have seen quite a number of data collection protocols proposed for WSNs. Some of them focused on the design either in the network layer [5, 6] or in the MAC layer [7]. To solve the common limited energy capacity issue in WSNs, they usually adopt or design an underlying MAC protocol with duty-cycled radio operations, so that the radio of each sensor node can wake up and sleep periodically based on an established time schedule. Thus the network energy efficiency can be improved by reducing two of the most principal energy wastage sources, i.e., idle listening and overhearing. However, the design separating the network and MAC layers may degrade the network performance due to the less concern on cross-layer cooperation. In addition, such duty-cycled radio operations will lead to a significant packet delivery latency, known as *sleep latency* [8], due to the fact that a sender has to hold its transmission until its receiver wakes up. Especially, in a multi-hop transmission, the single-hop latency will be accumulated, leading to a large end-to-end packet delivery latency. Therefore, the conventional duty-cycled schemes may not be able to meet the real-time delivery requirement of the delay-sensitive applications.

Recently, Linear Sensor Network (LSN) has attracted increasing attention for their promising application of monitoring a structure or area with a linear topology [9–12], e.g., oil/gas/water pipelines, railway/subway, roadway/highway, certain tourism/heritage sites, such as the Great Wall, etc. Due to the linear characteristic of the monitored structure, the nodes in an LSN are deployed in a linear form. This dissertation applies the proposed data collection protocol to an LSN, and model and analyze the performance of the LSN in terms of several performance metrics.

In addition, the past decades have also seen an increasing amount of attempts focusing on the analytical description of system characteristics and performance metrics through physical interference modeling and analysis for WANETs. As one of the promising tools, stochastic geometry [13] has been widely adopted, where the node distribution is assumed to follow a Poisson Point Process (PPP) [14–16] or a Binomial Point Process (BPP) [17–19]. However, the PPP model is inadequate/inaccurate in many practical wireless networks where a finite number of nodes are randomly distributed in a finite area, because it assumes an unbounded number of nodes and does not take into account the effects of the network boundaries. Meanwhile, the BPP

model can analyze neither the exterior interference at a reference receiver nor the average performance metrics at any node but only at a specific reference node. Both models provide us with average results over time and space, but cannot present performance metrics for a specific network deployment or a time instance [20]. Since most of the performance metrics in finite WANETs are nonlinear functions of the distances among communicating, relaying, and interfering nodes, this dissertation will also focus on the physical interference model based on distance, which has been extensively studied and applied as a significant complementary tool to the PPP and BPP models to quantify these metrics, such as interference [21], Signal-to-Interference-plus-Noise Ratio (SINR) [22], path loss [23], node degree [22], link/hop distance [24, 25], outage probability [23], link capacity [22], localization [26], energy consumption [27], stochastic properties of a random mobility model [28, 29], etc. As a result, Nodal Distance Distributions (NDD) is eventually involved in such quantifications, which intrinsically depend on the network coverage and nodal spatial distribution.

1.2 Research Objectives and Contributions

1.2.1 Data Collection Protocol Design

To alleviate the sleep latency issue, this dissertation adopts a pipelined scheduling scheme over duty-cycled radios. The basic idea is to stagger the sleep-wakeup schedules of the nodes along a forwarding path, so that a relaying node can forward a received packet in a short time, reducing the queuing time of the packet during the sleep periods and thus reducing the end-to-end packet delivery latency. In addition, the scheme can also mitigate the interference in the network, since any two communicating nodes will not be interfered with by their previous/next-hop neighbors (as they are sleeping according to their schedules). Due to the aforementioned features, the traffic contention in a congested area can also be handled effectively by moving data quickly away from that area.

Obviously, schedule synchronization plays a fundamental role in achieving such a pipelined scheduling design. There have been several local synchronization schemes proposed for non-duty-cycled radios [30–34]. As investigated in [35], for duty-cycled radio operations, however, these schemes may cause exponential error proliferation and impose a significant challenge on the scalability and efficacy of the synchronization. In addition, data collection and schedule synchronization should be considered

comprehensively, especially when the duty-cycled and pipelined scheduling features are involved. The separate designs of network and MAC layers may result in a high protocol overhead and inefficient usage of limited sensor node resources. Furthermore, the synchronization of duty-cycled, pipelined schedules along a forwarding path becomes quite subtle and challenging, since the schedule adjustment of any nodes along the path may lead to the out-of-sync with its previous/next-hop neighbors in a cascaded manner. Especially, due to the fact that different communicating nodes have different *clock drift velocities* in real hardware platforms and if not handled appropriately, the established pipelined schedules could be easily disrupted, which is referred to as *Duty-cycled, Pipelined Synchronization* (DPSync) issue and investigated by experimentation in Chapter 2.

To address the issues identified above, this dissertation proposes a Pipelined Data Collection (PDC) protocol for data collection in duty-cycled sensor networks. The whole protocol only relies on an RTS/CTS-like handshake, which is not only for data transmission as commonly utilized in prior work, but also for all other components in PDC, such as duty-cycled pipelined scheduling, schedule synchronization, data-gathering tree establishment, topology control and maintenance, etc. PDC integrates all the components of the protocol in a natural and seamless way to simplify the protocol implementation and to achieve a high energy efficiency and low packet delivery latency. PDC is then implemented in the latest Contiki Operating System (OS) [36] (a pioneer open-source OS for the Internet of Things) and a testbed is built based on two hardware platforms (Z1 [37] and MicaZ [38]) to evaluate the synchronization performance of PDC. The network performance of PDC is shown in comparison with a *de facto* standard for data collection based on the fully emulated Z1 in the Cooja simulator, which is provided by the Contiki OS for the rapid development of sensor networks.

PDC employs a fixed duty cycle, which, however, will cause unwanted energy consumption under light traffic loads, and network congestion and collision with packet retransmission or drop under heavy loads. This is because the accumulated data in the network cannot be sent promptly just in the active period of the radio, which further leads to a long packet delivery latency, low network capacity, and poor energy efficiency. Another concern about PDC is node addressing, e.g., the addressing for MAC/routing, which is often underestimated and even neglected in the prior data collection design. It is difficult and costly for the manufacturers of sensor nodes to assign a unique address for every node during the manufacturing phase, since there

are several issues to be considered carefully, such as address definition, address space management, address allocation, etc. Take the Z1 mote produced by Zolertia [37] for example. A single program in the Contiki OS has been provided to allow customers to manually assign a unique MAC address to each node one by one [39]. It is quite inconvenient, especially when there are a large number of nodes to be deployed. In addition, a WSN may need different sensor platforms from different manufactures due to their different sensing capabilities. Different manufacturers, however, may have different addressing schemes, which will obstruct the cross-platform communications in a heterogeneous network constituted of various sensor platforms. On the other hand, the execution of an independent address allocation and exchange mechanism in runtime also causes a significant network overhead. Note that in a dense WSN, it is quite difficult to link the address of a node for communication with its location, and thus people assume the required location information can be determined by other means (e.g., GPS) and embedded in the payload of a packet if necessary.

With the above considerations, this dissertation further proposes an Adaptive Data Collection (ADC) protocol based on PDC with two additional features integrated, i.e., free addressing and dynamic duty-cycling. Specifically, each node is identified for inter-node communication by using a Randomly-generated IDentifier (RID), plus its communication hop distance to the sink node. So there is no need to preassign a unique address to each node or perform an address allocation and management procedure in the runtime of a network. Furthermore, the sleeping period can be utilized on demand for data transmission to achieve a dynamic duty cycle adaptive to the network traffic load. Therefore, with the above two features, the network performance can be largely improved in terms of network heterogeneity, load adaptivity, and energy efficiency.

ADC has also been implemented in Contiki. A testbed based on two real hardware platform, i.e., Z1 and MicaZ, forming a heterogeneous network, and a Cooja simulated network constituted of several fully emulated platforms, such as Z1, Tmote Sky [40], and EXP5438 [41] are established to evaluate the performance of ADC. The evaluations are conducted in comparison with PDC, which demonstrates the practicality and efficacy of the design.

1.2.2 Protocol Modeling and Analysis

Being aware that, for LSNs, there is few work on the network modeling and analysis based on a duty-cycled MAC protocol, this dissertation applies a duty-cycled, pipelined-scheduling protocol (say, PDC proposed in this dissertation) to an LSN, models the protocol, and analyzes its performance in the LSN in terms of the system throughput, active time ratio per cycle of each node (which indicates the energy efficiency of the protocol), and packet delivery latency. The model consists of two parts. One is to analyze the MAC process, and the other is to model the queueing behavior with and without retransmission. Therefore, the model can be easily generalized to other duty-cycled protocols only by changing its first part to the analysis of the corresponding protocol process. All the sensor nodes in the model can operate as the source nodes to generate packets as in reality, i.e., the relaying nodes can also generate their own packets, which is one of the significant challenges facing the modeling of duty-cycled protocols in a multi-hop wireless network. To the best of our knowledge, none of existing models for a duty-cycled protocol consider this multi-source, multi-hop issue. Besides, the proposed model is validated under various scenarios in comparison with extensive simulations in OPNET, which is a well-known, industry-strength network simulator with high fidelity. In addition to enabling the effective estimation of the protocol performance, the proposed model and analysis provide an insightful understanding on the behavior of a duty-cycled protocol and aid its design and optimization for a multi-hop LSN.

1.2.3 Approach to Ran2Ran NDD

In the design and modeling of PDC above, the transmission and interference ranges are defined for successful communications between a pair of nodes. So PDC does not take into account the cumulative interference from the transmitters which are out of the contention range of a receiver. Since most performance metrics in wireless networks, such as outage probability, link capacity, etc., are nonlinear functions of the distances among communicating, relaying, and interfering nodes, a physical interference model based on distance is definitely needed in quantifying these metrics. Such quantifications eventually involve the NDD in a finite network intrinsically depending on the network coverage and nodal spatial distribution.

Using simulation can obtain NDD, which, however, is usually time-consuming and requires a large number of runs to fine-tune parameters to obtain statistically signif-

icant results. Moreover, by simulations, only the empirical Cumulative Distribution Function (CDF) of nodal distances can be obtained, while the accurate Probability Distribution Function (PDF) is indispensable to the modeling and analysis of wireless networks. Due to the significance of the path-loss exponent, even two close PDFs of nodal distances will lead to a substantial difference in performance metrics. Therefore, how to effectively obtain the relevant NDDs is definitely significant to accurately quantify the distance-dependent performance metrics when modeling and analyzing finite wireless networks, which has drawn plenty of attention in recent years [23, 42–59].

In general, there are two types of NDD involved in modeling and analyzing wireless communication networks, namely, 1) Ref2Ran NDD: the distribution of the distance between a given reference node (e.g., a base station in a cellular system) and a random node (a cellular user equipment), and 2) Ran2Ran NDD: the distribution of the distance between two random nodes. Recently, the Ref2Ran NDDs have been extended from the networks in certain specific shapes, including squares [42], disks/circles [23], hexagons [43, 47], regular polygons [54], and convex n-gons [53], where the reference node has to be inside or on the boundary of the network, to the networks in the shape of arbitrary polygons [58, 59], where the reference node could be anywhere.

In contrast, the Ran2Ran NDDs are still confined to the networks in certain specific shapes, including disks [21, 46, 51, 60], triangles [52, 55], rectangles [22, 24, 25, 51], rhombuses [44], trapezoids [50], and regular polygons [22, 27, 45, 48, 49, 56], which greatly limits the applicability of these approaches in modeling and analyzing wireless networks. For example, a practical finite wireless network may be in an arbitrary convex shape rather than the specific convex geometries mentioned above, or even in a concave shape; the network may contain two or more disjoint areas, or it has a tiered structure where two or more different networks are mixed deployed. To the best of our knowledge, there is lack of approach to the Ran2Ran NDDs associated with such a network, and thus the corresponding Ran2Ran NDD-based network modeling and analysis are infeasible.

This dissertation proposes a systematic and algorithmic approach to Ran2Ran NDDs by utilizing a concept, called Kinematic Measure (KM) in integral geometry [61], and decomposition and recursion methods. The approach can handle arbitrarily-shaped networks, including convex, concave, disjoint, and tiered networks, as well as different node densities in different network subareas, i.e., non-uniform nodal distributions. All the network shapes handled in the existing work mentioned

above are just the special cases in the proposed approach, which enables a wide range of new significant NDD-based network modeling and performance analysis. For example, the performance metrics in the networks with concave shapes and/or holes inside now can be accurately captured and analyzed, and the mutual influences among disjoint network regions (such as cells/clusters) or the networks mixed deployed in a tiered structure (e.g., a cellular system with underlaying device-to-device communications) can also be quantified properly. Based on the proposed approach, this dissertation derives and validates the Ran2Ran NDDs for the networks in the shapes of arbitrary polygons, as commonly seen in the current literature to approximate the shapes of finite wireless networks in the existing work. Specifically, the NDDs associated with arbitrary triangles are first obtained, which are further utilized to obtain the NDDs associated with arbitrary polygons, since any polygons can be triangulated.

1.2.4 Physical Interference Modeling and Analysis

Analyzing LSNs

With the proposed approach to Ran2Ran NDDs, this dissertation proposes a physical interference model framework to analyze the cumulative interference and link outage probability for an LSN running the PDC protocol. It can provide insights into tuning the protocol parameters, such as sleep factor, ξ, which determines the sleeping time of a node, provided with the outage probability of sensor nodes. There is an obvious tradeoff in setting ξ in PDC. A smaller ξ may lead to more concurrent transmissions, while the cumulative interference to the receiver of each communicating node pair increases as well, which more likely results in unsuccessful communications. The proposed physical interference modeling and analysis can help select a proper ξ to achieve an optimal network performance.

Analyzing Ad Hoc Device-to-Device Communications

The framework based on NDD is further applied in modeling and analyzing the ad hoc Device-to-Device (D2D) communications underlaying a cellular network. Recently, great efforts from both academia and industry have been devoted to the research and development of ad hoc D2D communications, as believed to be one of the promising technologies to improve network performance in several aspects [62]. Specifically, by allowing the direct communications between nearby User Equipments (UEs) with-

out traversing the Base Station (BS) or core network, not only are the transmission delay and power reduced, but also the network coverage can be extended. More importantly, D2D communications in cellular networks occurring on either the cellular or unlicensed/unused spectrum bring a great chance to improve network capacity and spectrum efficiency.

One of the serious challenges facing the implementation of D2D communications in cellular networks is the possible interference between D2D and cellular links and that between D2D links due to spectrum sharing. Considering an uplink resource reusing scenario, a Cellular UE (CUE) sends packets via the BS, while D2D UEs (DUEs) exchange packets directly in an ad hoc style utilizing the uplink radio resource. A D2D communication might be affected by a simultaneous transmission from CUE to the BS. In particular, as the number of concurrent D2D pairs increases, not only will the quality of the received signal at the BS be highly affected by the accumulated interference from the transmitting DUEs, but also the performance of each D2D pair itself will be degraded due to the inter-D2D interference. Likewise, in a downlink reusing scenario, the transmitting DUEs may make nearby CUEs fail to receive any signal. Therefore, the interference analysis in such a system is of great importance. An accurate interference analysis can provide us with deep insights into system performance in terms of several important performance metrics, such as SINR and those which are functions of SINR, including outage probability, capacity, etc.

In light of the significance of the interference analysis in cellular networks with D2D communications, extensive research has been conducted. Existing work mainly focused on either a simple scenario with single D2D pair [63–65] or throughput bound analysis [66, 67]. Recently, the tools from stochastic geometry have been adopted for the interference analysis of D2D communications in a large-scale cellular system with multiple cells working on the same frequency bands [68, 69]. However, these results cannot apply directly to a system with pre-deployed BSs according to cell planning, higher frequency reuse factors, or sector-partitioned cells. How to quantify the interference and system performance in more general scenarios is still an open issue, inspiring the work in this dissertation as a significant complementary one to the existing approaches and results.

Different from the existing work, this dissertation presents a framework based on both Ref2Ran and Ran2Ran NDDs with a path-loss model to obtain the distributions of signal, interference and further SINR, which are further utilized to thoroughly investigate the interference and outage probabilities for both cellular and D2D com-

munications. Since uplink resources are more likely to be shared than downlink resources as a result of the asymmetric uplink and downlink service loads [70, 71], under-utilization of uplink spectrum [72], and stronger abilities at BSs to process interference than at CUEs [71], the framework focuses on an uplink reusing scenario, where multiple DUEs reuse the uplink resource of a CUE. Specifically, according to a general path-loss model commonly adopted for analyzing wireless networks, the signal/interference received at a node depends heavily on the distances to its transmitting/interfering nodes. Considering that the UEs are independently and uniformly distributed in a cell, we first obtain the distributions of the distance from the BS to a random UE and that between two random UEs, associated with arbitrarily-shaped service areas. The technique can be extended to non-uniformly distributed UEs. The case that a Guard Region (GR) is set for the BS to mitigate the interference from DUEs is also considered, where DUEs are randomly deployed beyond the GR. Meanwhile, the shadowing and fading effects can also be easily included, as shown in Section 5.4.5, and the proposed framework can also be applied to a downlink reusing scenario, as detailed in Section 5.4.4. The proposed framework has no limitations on the network shapes, and the presented results can provide important insights and guidelines for network planning and dimensioning at the system level.

1.3 Dissertation Organization

This dissertation covers topics in protocol modeling and performance evaluation in WANETs. The remainder of the dissertation is organized as follows.

In Chapter 2, we investigate the duty-cycled, pipelined schedule synchronization issue by experimentation, and illustrate that the established pipelined schedules over duty-cycled radios along a forwarding path can be easily disrupted if the issue is not handled appropriately. Then we propose PDC with a cross-layer integration of all the protocol components in a natural and seamless way. ADC is further proposed based on PDC to take into account free addressing and dynamic duty-cycling. Both PDC and ADC are implemented in the Contiki OS and a testbed is built to evaluate their performance.

In Chapter 3, we model a duty-cycled protocol with a pipelined-scheduling feature, i.e., the PDC protocol proposed in Chapter 2, for an LSN. Based on the model, we analyze the network performance and validate the model through OPNET simulations.

In Chapter 4, a systematic and algorithmic approach is proposed to obtain Ran2Ran NDDs. Based on the proposed approach, we derive and validate the Ran2Ran NDDs for the networks in the shapes of arbitrary polygons, as commonly seen in the current literature to approximate the shapes of finite wireless networks.

In Chapter 5, a physical interference model framework based on NDDs is proposed to analyze the LSN running the PDC protocol, so the cumulative interference at a receiver can be considered. The framework is further applied to analyze ad hoc D2D communications underlaying a cellular network.

Chapter 6 concludes the dissertation and discusses the future work.

1.4 Bibliographic Notes

Most of the works reported in this dissertation have appeared in research papers. The work on PDC in Chapter 2 has been published in [73, 74] and that on ADC has appeared in [75] (an extension of which has been submitted as [76]). The works in Chapter 3 have been published in [77, 78]. The works in Chapter 4 have been briefly announced in [57, 79]. Some parts in Chapter 5 have appeared in [80, 81].

Chapter 2

Energy-Efficient and Practical Pipelined Data Collection

2.1 Overview

In this chapter, we first investigate the synchronization issue facing the duty-cycled, pipelined data collection in WSNs. Then we propose a Pipelined Data Collection (PDC) protocol, which takes into account both the pipelined data collection and the underlying schedule synchronization over duty-cycled radios practically and comprehensively. Based on PDC, we further propose an Adaptive Data Collection (ADC) protocol, with two additional features naturally and seamlessly integrated, i.e., free addressing and dynamic duty-cycling, to improve network heterogeneity, load adaptivity, and energy efficiency.

2.2 Related Work

In this section, we review the existing work on pipelined scheduling, schedule synchronization, dynamic duty-cycling, and free addressing in duty-cycled sensor networks, respectively.

2.2.1 Pipelined Scheduling with Duty Cycling

Forwarding data packets along a multi-hop path in a pipelined fashion to reduce the packet delivery latency could be found in some of the existing duty-cycled MAC protocols designed for sensor networks. These MAC protocols can be classified into

two categories: with and without an integrated routing scheme. For the latter, for example, Cao et al. [82] proposed a streamlined wake-up algorithm to organize the duty cycles of nodes into a streamlined sequence to forward data promptly. Du et al. [83] proposed a duty-cycled MAC protocol, which awakens part of the nodes along a forwarding path sequentially in a single operation cycle, so that a packet can be delivered over multiple hops in one cycle. Overall, these protocols only focused on the design in the MAC layer, and had no consideration on schedule synchronization, which is, however, indispensable to achieve the pipelined forwarding.

There are also a set of duty-cycled MAC protocols integrating routing over a pre-constructed data-gathering tree, such as DMAC [84], MERLIN [85], PRI-MAC [73], etc. However, these protocols either ignored the synchronization issue, or just assumed that a local synchronization scheme for non-duty-cycled sensor networks could meet the requirement of achieving the duty-cycled pipelined scheduling. For example, DMAC based on a tree structure assumed that a local synchronization is enough and some of the existing schemes such as RBS [30] could meet its requirements, without evaluating their efficacy and mutual effect with DMAC on the network performance. In addition, DMAC provided no details about how to build the tree and how to stagger sleep-wakeup schedules among sensor nodes. MERLIN and PRI-MAC had no consideration on schedule synchronization, either. Furthermore, all these protocols were evaluated only through simulations. While on real hardware platforms, the performance of a network running these protocols may degrade tremendously due to the out-of-sync issue, since there is no practical and appropriate synchronization scheme provided.

Recently, Cao et al. [8] proposed a Robust Multi-pipeline Scheduling (RMS) algorithm for data collection in duty-cycled sensor networks. In RMS, multiple parallel pipelines are coordinated so that a packet can be switched timely among different pipelines if failure happens during its former attempts of transmission. By combining the pipelined scheduling scheme and the multi-parent forwarding scheme, RMS could minimize the end-to-end delivery latency while handling unreliable wireless communication links and transmission failures. RMS did not consider schedule synchronization, either, but just assumed that the clocks on each sensor in the neighborhood can be locally synchronized by utilizing FTSP [31], a flooding time synchronization for non-duty-cycled sensor networks. However, as we will show in Section 2.3, the DPSync issue cannot be eliminated with a local synchronization scheme. In [86], a circular pipelining algorithm was proposed to reduce the end-to-end delivery latency

for round-trip network operations in duty-cycled sensor networks. Similarly, the work also assumed that the network is locally synchronized by using FTSP.

2.2.2 Schedule Synchronization with Duty Cycling

Schedule synchronization is essential to various sensor network operations, especially for duty-cycled sensor networks, as a node can communicate with any one-hop neighbor only when both of them are active with radios on. The intermittent link due to the duty-cycled radio operations makes wireless communications in sensor networks very challenging with the practical issue of clock drift, which in turn makes the synchronization itself quite difficult. There have been several synchronization protocols proposed for non-duty-cycled sensor networks, such as RBS [30], FTSP [31], FBS [32], and MTS [33, 34], by utilizing the broadcast channel [30] and multi-point time-stamping with linear regression [31], respectively. However, these methods are not suitable for a duty-cycled scenario and can cause exponential error proliferation [35]. Especially, for a duty-cycled, pipelined data collection, these methods are faced with the DPSync issue, which has not been addressed in the current literature.

On the other hand, PSR [87] is one of the earliest work on synchronous rendezvous in duty-cycled sensor networks. The key idea in PSR is to extract timing information embedded in the duty-cycled pattern of radios by utilizing the normal traffic in the network, which can reduce the time-stamp exchange overhead with dedicated packets in the traditional methods for non-duty-cycled networks. Since PSR was proposed as a generic element only at the MAC layer without taking into account the network topology and specific applications, it could not be utilized to meet the requirement for achieving multi-hop pipelined forwarding. LDSP proposed in [35] is another synchronization protocol for duty-cycled wireless networks. Similar to PSR, it focuses on the local synchronization of a pair of nodes. With the proposed parallel synchronization mechanism, LDSP could effectively achieve the synchronization of an ordinary node with a global reference node. Nevertheless, it is not applicable for duty-cycled, pipelined data collection due to the same reason as PSR.

2.2.3 Dynamic Duty-Cycling

Dynamic duty-cycling could be found in some of the existing duty-cycled MAC protocols designed for sensor networks [84, 88–91]. For example, DMAC [84] based on a tree structure could utilize the sleeping period for data transmission on demand.

But DMAC only implemented through simulation did not consider the duty-cycled, pipelined synchronization issue, and provided no details about how to build the tree and sleep-wakeup schedules among sensor nodes. In [88], a node adaptively determines its radio schedules based on the traffic patterns of its own and neighbors; Wang et al. [89] focused on the end-to-end delay guarantees with a dynamic duty-cycle control approach; Zhang et al. [90] proposed a localized and on-demand duty cycling scheme, which allows a node to adaptively adjust its duty cycle; and Aby et al. [91] proposed an adaptive MAC protocol using the history information of successful frame exchanges to compute the next activation times. Overall, these protocols were not designed in particular for data collection, where the many-to-one traffic pattern was not considered. Meanwhile, only focusing on the performance evaluation of the MAC protocol, it remains to be seen the whole network performance when working with an independent network-layer (routing) protocol.

2.2.4 Free Addressing

Due to the addressing issues in WSNs as identified in Chapter 1, as well as in [92–96], the idea for free addressing has appeared in the existing literature at its beginning for non-duty-cycled sensor networks [92–94, 96]. Known as the first work which introduced the concept of "address-free", Elson et al. [92] proposed an address-free architecture for a WSN, where nodes randomly select probabilistically unique identifiers. However, the proposed architecture could not guarantee the absence of identifier conflicts and the reliability of data transmission. In a data collection scenario, Jobin et al. [93], Chen et al. [94], and Fang et al. [96] eliminated the addressing issue by allowing a node to forward its data to any of its neighbor nodes closer to the sink node, and thus there is no need of unique addresses. However, for duty-cycled, pipelined data collection, these schemes are inapplicable, since a node with pending data has to identify its forwarder which it keeps in sync with. To the best of our knowledge, [95] is the first work which applied a free-addressing scheme in duty-cycled data collection of sensor networks, by generating a short random identifier for each new transmission of a node. However, the proposed scheme was designed only for detecting rare events with a fixed duty cycle, which may cause a high packet delivery latency, and only implemented in simulation. Different from the above schemes, ADC proposed in this dissertation will have a comprehensive consideration on free addressing for a pipelined data collection with dynamic duty-cycling.

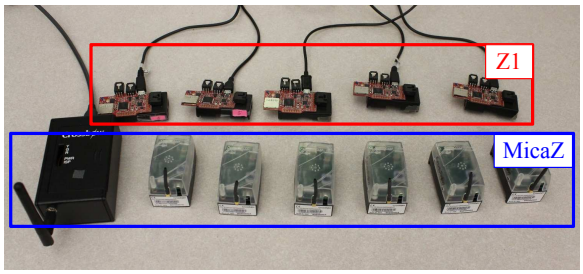


Figure 2.1: The two hardware platforms, Z1 and MicaZ, used in our testbed implementation for protocol performance evaluation.

Table 2.1: Z1 and MicaZ features

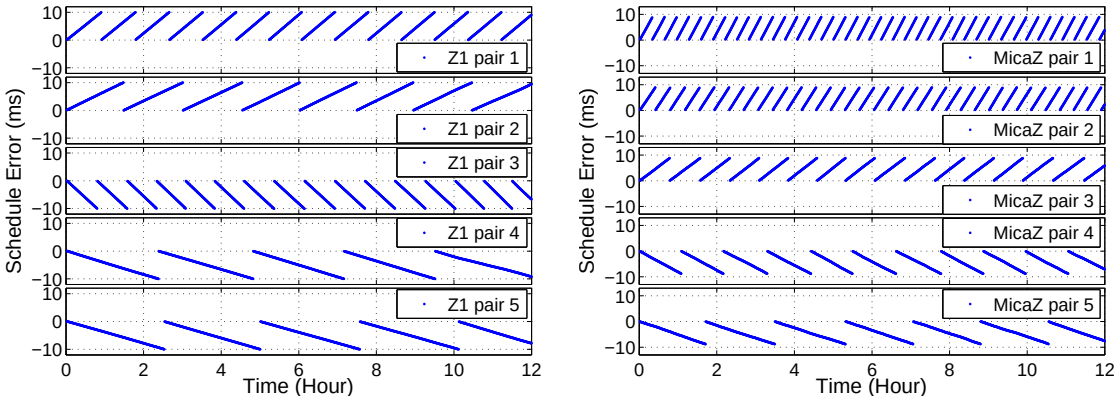
	Frequency Range	**Processor**	**Radio Transceiver**	**Flash**	**RAM**
Z1	2.4 GHz	TI MSP430	TI CC2420	92 KB	8 KB
MicaZ	2.4 GHz	Atmel AVR	TI CC2420	512 KB	4 KB

2.3 Duty-Cycled Pipelined Synchronization Issue

With the existing synchronization mechanisms, the duty-cycled, pipelined data collection in WSNs is faced with the DPSync issue. In this section, we first present and analyze the experiment results on the schedule error between two motes under duty-cycled radio operations based on our testbed. The testbed consists of two hardware platforms, Z1 [37] and MicaZ, as shown in Figure 2.1 with their features listed in Table 2.1. Then we investigate the DPSync issue in detail.

2.3.1 Schedule Error Measurement and Analysis

To deeply understand the duty-cycled schedule error due to factors such as hardware and OS latency and clock drift, and provide insights into the practical schedule synchronization in duty-cycled sensor networks, we have done experiments with ten different, arbitrarily chosen pairs of Z1 motes and ten pairs of MicaZ motes. For each pair, one of the motes is denoted as the *reference* mote, the other is the *observing* mote which will measure the schedule error between itself and the reference mote through a packet handshake with time-stamps embedded. Specifically, the observing mote sends a packet, say P_1 to the reference mote, and the reference mote replies with P_2 after receiving P_1. So the two motes are synchronized with each other at the beginning. Then every 30 sec, the observing mote measures the schedule errors rela-



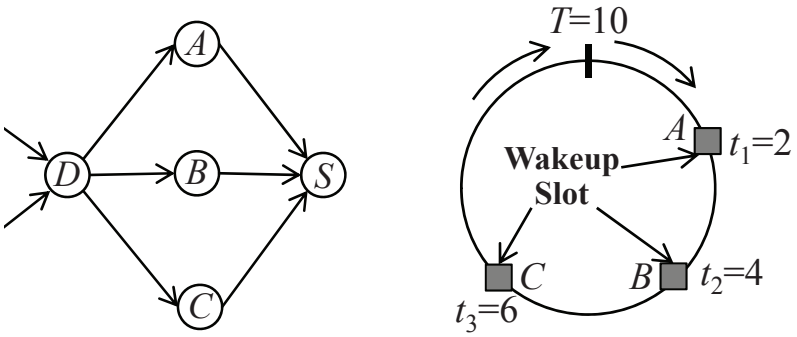
(a) Schedule error between two Z1 motes (five different, arbitrarily chosen pairs)

(b) Schedule error between two MicaZ motes (five different, arbitrarily chosen pairs)

Figure 2.2: The schedule error between two motes over time. When the error reaches ten ms, the *observing* mote adjusts its schedule to keep in sync with the *reference*, so that the error between them will not be increased endlessly.

tive to the reference mote through the P_1/P_2 handshake. The observing mote adjusts its clock to synchronize with the reference mote when the error reaches ten ms, so that the schedule error between them will not be increased endlessly. All experiments lasted for more than 24 hours. For the ease of presentation, Figure 2.2 only shows the schedule errors of five Z1 and five MicaZ pairs over 12 hours, with similar results observed for other pairs and over the entire time duration.

Figure 2.2 shows that for each pair, the schedule errors measured at the observing mote have a linear accumulation. This is reasonable as the the two motes were placed under the same and stable indoor environment and thus the clock skew between two clocks almost has no change. Figure 2.2 also shows that, for different pairs, the schedule errors could be either positive or negative, i.e., they have different *clock drift directions*. For example, for Z1 pairs 1 and 2 shown in Figure 2.2a, and MicaZ pairs 1, 2 and 3 shown in Figure 2.2b, the schedule errors are positive, while for Z1 pairs 3, 4 and 5 and MicaZ pairs 4 and 5, the schedule errors are negative. In addition, it is obvious that the clock skew, which indicates the instantaneous *clock drift rate* between two clocks [97], is different for different pairs. Especially, for the same pair of motes, in a practical outdoor deployment, the clock skew between the two motes may change variously, and not necessarily have no change as in our indoor experiment. Based on the above observations, i.e., different pairs having different *clock drift velocities* (including different directions and rates), the DPSync issue is analyzed below.



(a) An example topology

(b) Multi-pipeline schedules maintained by node D

Figure 2.3: Illustration of the DPSync issue.

2.3.2 DPSync Issue Analysis

Most of the previous duty-cycled, pipelined scheduling schemes neglected or underestimated the DPSync issue by assuming that existing synchronization schemes for non-duty-cycled scenarios could meet the fundamental requirements in their designs [8, 73, 84–86]. Figure 2.3a shows a simple example where D has three relaying nodes and there are three forwarding paths, i.e., $D \rightarrow A \rightarrow S$, $D \rightarrow B \rightarrow S$, and $D \rightarrow C \rightarrow S$. S provides a reference clock. The one-hop away nodes (A, B and C) adjust their clock when necessary to synchronize with S. Such an adjustment of any of the three nodes will lead to the clock adjustment at D to synchronize with them, which further leads to the clock adjustment at D's previous-hop senders, and so on and so forth. However, such a synchronization in a cascaded manner can easily disrupt the established pipelined scheduling if not handled appropriately, as analyzed below in detail.

In [73, 84, 85], the nodes located the same hops away from the sink node will have the same schedule, while in [8] and [86], they will have different schedules to achieve multi-pipeline scheduling. Each case is analyzed below by taking Figure 2.3a as an example:

Case I: *A*, *B* and *C* have the same schedule. Therefore, D only needs to maintain one schedule after synchronizing with A, B and C. Every time when D has a packet to send or forward, it can choose one of them as the next hop. However, this is not the case in reality. First, the schedules maintained by A, B and C will become increasingly different due to their different clock drift velocities relative to S. Second, due to the different clock drift velocities at D relative to different nodes,

i.e., A , B and C, node D can hardly maintain only one schedule to synchronize with all the three nodes. Then the original assumptions and designs in the previous work could not be achieved. In addition, even if D could maintain different schedules for all of its next hops, all the nodes having D as the next hop need to maintain multiple schedules as well, and so forth. With multiple schedules maintained at each hop and the utilization of existing schemes for local synchronization, it is hard to achieve multi-hop pipelined scheduling.

Case II: *A*, *B* and *C* have different schedules. As in [8], we use a circle to represent the time line of each work period. Assume that with a schedule cycle of $T = 10$, the schedules of node A, B and C after synchronizing with S are $\{2\}$, $\{4\}$ and $\{6\}$, respectively. To achieve multi-pipeline scheduling, node D maintains its own schedule according to these schedules, as shown in Figure 2.3b, and all the nodes having D as their relaying node maintain their schedules according to D's schedule, and so on and so forth [8]. However, this is also not the case in reality, due to the same reasons above, i.e., different clock drift velocities for different pairs. Suppose that the clock drift velocities at node A, B and C relative to S are 0.3, 0.1 and -0.1, respectively. Then after one cycle ($T = 10$), all the three nodes have the same schedule, i.e., $\{5\}$. If they can still communicate with S, they do not need to adjust their clocks to synchronize with it. As a result, the multi-pipeline schedules along the paths involving D could be disrupted.

For both of the cases, the clock adjustment for synchronization at a node will lead to the clock adjustments of all its children and descendants in the pipelined-forwarding paths, which, however, is not handled appropriately in the existing schemes for local synchronization, and thus makes the established pipelined scheduling fragile and impractical.

2.4 Pipelined Data Collection

This section presents the design of PDC in detail. We first have an overview of PDC in Section 2.4.1. Then Sections 2.4.2 and 2.4.3 present a set of algorithms for PDC to achieve pipelined scheduling over duty-cycled radios and build the data-gathering tree, respectively. A practical, effective duty-cycled schedule synchronization scheme naturally incorporated in PDC is presented in Section 2.4.4. Section 2.4.5 discusses the features in PDC related to topology control and maintenance, including easy network deployment, resilience to sync failure, and adaptive response to topology

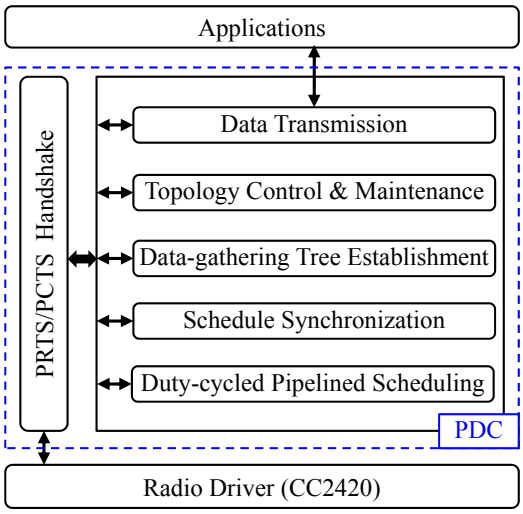


Figure 2.4: The implementation architecture of PDC in the Contiki OS. Only relying on the PRTS/PCTS handshake, all the other components in PDC are naturally integrated and able to support each other.

changes. Finally, Section 2.4.6 shows the implementation and evaluation of PDC.

2.4.1 Design Overview

PDC has taken into account the issues identified above carefully and comprehensively, such as energy consumption, sleep latency, pipelined scheduling, schedule synchronization, topology control and maintenance, etc. In PDC, data collection and duty-cycled MAC are naturally integrated through a cross-layer integration design to reduce the overhead of network communications, and to keep the superiority of the duty-cycling schemes to increase energy efficiency. The nodes along a path from its source node to the sink node have staggered sleep-wakeup schedules and thus data can be forwarded in a pipelined fashion, largely reducing the packet delivery latency and efficiently handling the traffic congestion by moving traffic quickly away from the congested area. By only relying on an RTS/CTS-like handshake, called PRTS/PCTS, PDC can achieve the above design goals and features effectively.

Specifically, to achieve duty-cycled pipelined scheduling, we propose an algorithm called Grade Division and Pipelined Scheduling (GDPS) for network initialization, as well as topology control and maintenance. GDPS divides all nodes into different grades (equivalent to their communication hop distances to the sink node) to establish pipelined schedules among them. The sink node in grade zero determines its

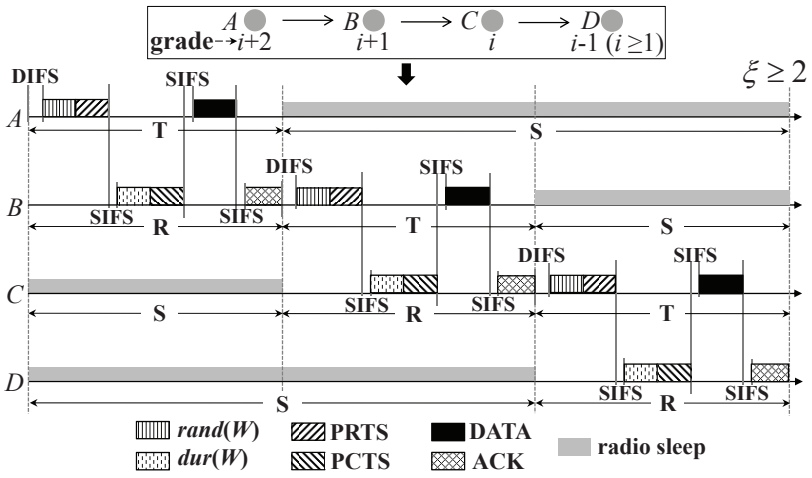


Figure 2.5: Data transmission in PDC. $rand(W)$: a random time duration within W; $dur(W)$: a function for calculating the waiting time for the current node.

schedule by itself and initiates GDPS by periodically broadcasting the PRTS packet. Other nodes keep the radio on to listen to the channel until they join a grade and thus determine their periodic sleep-wakeup schedules by parsing the received PRTS packets (all the other control or data packets of PDC they receive can be utilized for this purpose as well). Then these nodes also periodically broadcast PRTS, so that 1) those nodes who have not joined the network can join (as long as they are not isolated in the network), and 2) each node can find its parent node after receiving a PCTS reply from its adjacent lower-grade node, and finally a data gathering tree can be established and maintained. Once a node finds its parent, it starts the PRTS/PCTS unicast handshake either periodically for schedule synchronization as well as for topology control and maintenance, or immediately for data transmission (schedule synchronization, and topology control and maintenance can also be handled accordingly), depending on whether it has pending data to be transmitted or not. There are no other control packets necessary and all the components with two layers (i.e., MAC and network) incorporated are naturally and seamlessly integrated together and able to support each other, which can also simplify the implementation of PDC. The details of these components will be introduced in the following subsections. Figure 2.4 shows the implementation architecture of PDC in the latest Contiki OS (i.e., Contiki 3.0).

Assume a node has joined grade i ($i \geq 1$) and determined its sleep-wakeup schedule during the GDPS process. Then the node will experience periodically three consecutive states, namely, `R`: to receive a packet from one of its senders (children) in grade

$i+1$, `T`: to transmit a packet to its receiver (parent) in grade $i-1$, and `S`: to sleep. The schedules between two adjacent grades are staggered so that after a node receives a data during its `R` state, it can forward the data immediately during its upcoming `T` state. Figure 2.5 shows an example about the data transmission along a path $A \rightarrow B \rightarrow C \rightarrow D$.

Besides handling the hidden terminal problem, it is worth noting that the PRTS/PCTS handshake in PDC differs in three aspects from the traditional RTS/CTS handshake in the IEEE 802.11 MAC (also widely utilized in some of the existing well-known duty-cycled MAC protocols such as S-MAC [98] and T-MAC [99]). First, the PRTS sent by a node contains the grade information of the node. Only those adjacent lower-grade nodes will reply it with a PCTS. Second, if a node has not found its parent node, the PRTS will be broadcast without containing the next-hop address. Those adjacent lower-grade nodes which simultaneously receive this PRTS will contend for replying with PCTS, and the node that wins the contention will become the next hop. Therefore, a node also needs to wait a time duration (i.e., $dur(W)$ in Figure 2.5) before replying with PCTS after receiving a PRTS. $dur(\cdot)$ is a function for calculating the waiting time, and W is the maximum number of mini time-slots allowed in the contention window of the handshake. $dur(\cdot)$ also ensures that those nodes with fewer child nodes reply first, which will be introduced in detail in Section 2.4.3. Third, as introduced above and shown in Figure 2.4, the handshake is also used for all the other components in PDC, but not just for data transmission.

In PDC, `T` is equal to `R`, which is the slot accommodating at most one transmission of one data packet with the maximum size predetermined by a specific application. Therefore, a normal slot duration in PDC is

$$\begin{aligned} T_{slot} &= 2W\sigma + \mathrm{DIFS} + 3\mathrm{SIFS} + durPRTS \\ &\quad + durPCTS + durDATA + durACK\ , \end{aligned} \tag{2.1}$$

where σ is the time duration of one mini time-slot in the contention window, and $durPRTS$, $durPCTS$, $durDATA$ and $durACK$ are the one-hop transmission durations of packets PRTS, PCTS, DATA and ACK, respectively. To achieve pipelined scheduling, the duration of `S` must be an integer multiple of T_{slot}, which is $T_S = \xi \cdot T_{slot}$, where the positive integer ξ is called sleep factor. So the whole cycle duration is

$$T_{cycle} = (\xi + 2) \cdot T_{slot}. \tag{2.2}$$

Table 2.2: PRTS Format

Fields	Size	Comments
grade	1 Byte	The grade X[1] belongs to.
state_dur	2 Bytes	The duration that X has stayed in state `T`.
source	2 Bytes	Address of node X.
next_hop	2 Bytes	Address of the target node. Set to *NULL* if the PRTS is to be broadcast.
not_for_data	1 Byte[2]	A flag, set if PRTS is not for data transmission.
child_num	1 Byte	If the PRTS is to be broadcast, this field is used to build the data-gathering tree; otherwise, this field indicates the number of children X's parent has.
schedule_error	2 Bytes	The schedule error between X and its parent, used when the PRTS is to be unicast. After receiving the PRTS, the parent will store it in CT.

[1] Assume the PRTS is sent by node X.
[2] Note that one bit is enough for this flag, which can be encapsulated with other flags (if any) in the physical layer.

Considering that interference is about twice the range of transmission for the wireless communication in an open-space environment [100], ξ needs to be at least two, so that the communication between two nodes will not be interfered with by their previous/next-hop neighbors. For example, in Figure 2.5, with $\xi \geq 2$, C and D in `S` state will not interfere with the communication between A and B. On the other hand, A and B in `S` state will not interfere with the communication between C and D. For setting an appropriate ξ, obviously, there is a tradeoff between energy efficiency (the larger ξ the better) and packet delivery latency (the smaller ξ the better), which depends on the requirement of a specific application. In Section 2.5, we will consider a dynamic duty-cycle design, i.e., by utilizing the sleep period on demand for data transmission and thus a higher energy efficiency and a shorter sleep latency could be achieved.

2.4.2 Grade Division and Pipelined Scheduling

In PDC, each node maintains three attributes, namely, 1) *grade*, the communication hop distance to the sink node, initialized to -1, 2) *state*, the current radio state of the node, which is in {`R`, `T`, `S`}, and 3) *state_dur*, the duration that the node radio has stayed in the current state. For GDPS, the sink node first sets its grade to zero and starts its periodic sleep-wakeup schedule. Then it broadcasts PRT-

Table 2.3: PCTS Format

Fields	**Size**	**Comments**
grade	1 Byte	The grade Y[1] belongs to.
state_dur	2 Bytes	The duration that Y has stayed in state R.
source	2 Bytes	Address of node Y.
next_hop	2 Bytes	Address of the target node.
child_num	1 Byte	The number of children Y has.

[1] Assume the PCTS is sent by node Y.

S as a periodic beacon in its T state (periodic beacon is typically used for topology control and maintenance in most of the existing realistic implementations of sensor networks [5, 101, 102]).

The packet format of PRTS is shown in Table 2.2. With the fields *grade* and *state_dur* in a PRTS set to the two corresponding attributes of the sink node and other fields set appropriately, the sink node broadcasts this PRTS after it enters T state and waits for a time duration, $DIFS + rand(W)$. A random time duration within W, i.e., $rand(W)$, is utilized to handle multi-sink/node channel contention (for easy and efficient management and maintenance, it is possible for PDC to support multiple sinks as roots for data collection, so a large-scale network can be divided into several subnetworks, which will be investigated in our future work).

A node which has not joined the network keeps its radio on to listen to the channel. It will set its grade and schedule to join the network by parsing a received packet (which can be any packet in PDC). To ensure that the node will choose the lowest grade to join, the node sets its final grade and schedule after receiving more than two packets. Then it will periodically broadcast a PRTS packet in its T state with the fields *grade* and *state_dur* set to its two corresponding attributes. It may happen that there is a collision since more than one node broadcasts PRTS simultaneously. Then these nodes will wait a random number of operational cycles to attempt another channel access. When the collision is eliminated, each node will continue the previous periodic PRTS broadcast, and it is highly possible that the broadcast orderings among them have drifted apart.

Figure 2.6 shows an example to illustrate GDPS. Suppose node B has joined grade i ($i \geq 1$) and established its schedule, and it sends PRTS either periodically or for data transmission in state T. Node A within the transmission range of B has not joined the network and keeps its radio on. It will parse the overheard packet sent from B to determine the three attributes to be maintained, i.e., *grade*, *state* and

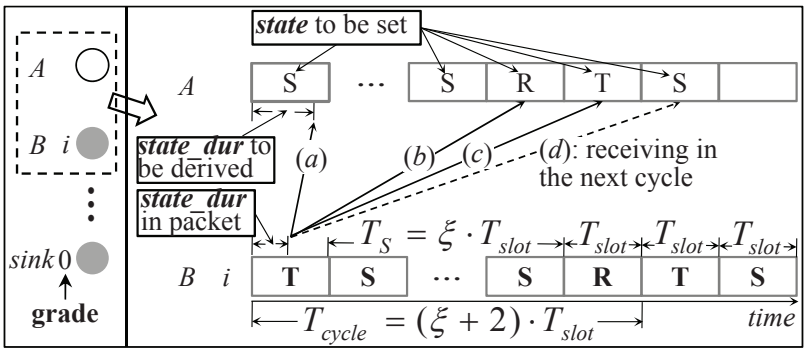


Figure 2.6: GDPS. For cases (a), (b) and (c), A receives PRTS from B in the current cycle, while (d) indicates that it receives a packet in the next cycle (a node in gray means that it has joined the network with its grade and schedule identified, while a node in white has not).

state_dur. Suppose A receives a packet from B which is in state `T` (the received packet by A could be either PRTS or DATA if B is sending data to its parent node). It is obvious that A will set its *grade* to $i+1$ according to the received packet. Depending on the packet propagation delay, denoted by T_p, which is the difference between the packet reception time and its generation time, when A receives the packet, it could be in either states `S`, `R` or `T`, corresponding to the cases (a), (b) and (c) shown in Figure 2.6, respectively. If T_p is long enough so that

$$\mathcal{K} = \left\lfloor \frac{\text{PRTS}.state\_dur + T_p}{T_{cycle}} \right\rfloor \geq 1 \ , \tag{2.3}$$

A will receive the PRTS after one (case (d)) or even more cycles. According to the periodicity of the schedule, the following calculation is to confine the set of *state* and *state_dur* in A to the three cases (i.e., cases (a), (b) and (c) shown in Figure 2.6) in one cycle,

$$t = \text{PRTS}.state\_dur + T_p - \mathcal{K} \cdot T_{cycle} \ . \tag{2.4}$$

The details of the three cases are

- case (a): $t < \xi \cdot T_{slot}$. Node A should set its *state* to `S` and its *state_dur* to t. After $(T_S - state\_dur)$ time, A enters the next state, `R`. If $\mathcal{K} = 1$, this is case (d) in Figure 2.6.

- case (b): $\xi \cdot T_{slot} \leq t < (\xi + 1) \cdot T_{slot}$. Node A has transitioned to state `R` from `S`. So A will set its *state* to `R` and its *state_dur* to $(t - \xi \cdot T_{slot})$. After

Algorithm 1 GDPS: parsing a received packet, P

1: $t \leftarrow \mathrm{P}.state\_dur + T_p - \left\lfloor \frac{\mathrm{P}.state\_dur + T_p}{T_{cycle}} \right\rfloor \cdot T_{cycle}$
2: **if** $grade < 0$ **or** $grade > \mathrm{P}.grade + 1$ **then**
3: $\quad grade \leftarrow \mathrm{P}.grade + 1$
4: $\quad$ **if** P is PRTS or DATA **then**
5: $\quad\quad$ **if** $t < \xi \cdot T_{slot}$ **then**
6: $\quad\quad\quad state \leftarrow$ `S`
7: $\quad\quad\quad state\_dur \leftarrow t$
8: $\quad\quad$ **else if** $t < (\xi + 1) \cdot T_{slot}$ **then**
9: $\quad\quad\quad state \leftarrow$ `R`
10: $\quad\quad\quad state\_dur \leftarrow t - \xi \cdot T_{slot}$
11: $\quad\quad$ **else**
12: $\quad\quad\quad state \leftarrow$ `T`
13: $\quad\quad\quad state\_dur \leftarrow t - (\xi + 1) \cdot T_{slot}$
14: $\quad\quad$ **end if**
15: $\quad$ **end if**
16: $\quad$ **if** P is PCTS or ACK **then**
17: $\quad\quad$ **if** $t < T_{slot}$ **then**
18: $\quad\quad\quad state \leftarrow$ `T`
19: $\quad\quad\quad state\_dur \leftarrow t$
20: $\quad\quad$ **else if** $t < (\xi + 1) \cdot T_{slot}$ **then**
21: $\quad\quad\quad state \leftarrow$ `S`
22: $\quad\quad\quad state\_dur \leftarrow t - T_{slot}$
23: $\quad\quad$ **else**
24: $\quad\quad\quad state \leftarrow$ `R`
25: $\quad\quad\quad state\_dur \leftarrow t - (\xi + 1) \cdot T_{slot}$
26: $\quad\quad$ **end if**
27: $\quad$ **end if**
28: **end if**

$(T_{slot} - state\_dur)$ time, it enters its next state, `T`.

- case (c): $(\xi+1) \cdot T_{slot} \leq t < T_{cycle}$. Node A will set its *state* to `T` and its *state_dur* to $(t - (\xi + 1) \cdot T_{slot})$. After $(T_{slot} - state\_dur)$ time, it enters the next state, `S`.

Note that when data transmission starts after the network is initialized, apart from receiving a PRTS or DATA packets from a node in state `T`, a newly added node can also receive PCTS or ACK packets from a node in state `R`. By choosing the lowest grade to join, it can also set its state and schedule by parsing PCTS or ACK packet based on GDPS. The PCTS packet format is shown in Table 2.3. Both DATA and ACK should contain at least two fields, *grade* and *state_dur* to support GDPS. The

Table 2.4: Parent Table (PT) at node X

Fields	**Size**	**Comments**
parent	2 Bytes	Parent address.
no_pcts_counter	1 Byte	The number of times that there is no PCTS received after unicasting PRTS.
schedule_error	2 Bytes	The schedule error between X and the *parent*.
child_num	1 Byte	The number of children the *parent* has.

Table 2.5: Child Table (CT) at node Y

Fields	**Size**	**Comments**
child	2 Bytes	Child address.
no_prts_counter	1 Byte	The number of times that there is no PRTS which is unicast to Y.
schedule_error	2 Bytes	The schedule error between Y and the *child*. Updated by the *schedule_error* of the received PRTS.

complete algorithm for parsing a received packet in GDPS is shown in Algorithm 1.

2.4.3 Data-Gathering Tree Establishment and Maintenance

For data-gathering tree establishment and maintenance, each node will maintain two neighbor tables, which are Parent Table (PT) and Child Table (CT). The table formats are shown in Table 2.4 and Table 2.5, respectively. PT has at most one item to record the parent node of the current node. CT is used to record the multiple child nodes of the current node. Once a node joins a grade and determines its sleep-wakeup schedule, it will periodically broadcast PRTS and wait in state `T` for a possible PCTS reply from an adjacent lower-grade node to construct its PT, which is called PRTS/PCTS broadcast handshake. Once PT is constructed, the node will change to periodically unicast PRTS to the parent recorded by its PT for schedule synchronization and topology control and maintenance, and meanwhile it can start PRTS unicast for data transmission at any time when there is a pending data packet to be sent, which is called PRTS/PCTS unicast handshake.

Grade-1 nodes can only have the sink as their parent node, while a node with $grade \geq 2$ could have more than one neighbor node (within its communication range) in its adjacent lower grade. Among all these neighbor nodes, the node will choose the

Algorithm 2 Periodic PRTS broadcast in state `T` for data-gathering tree establishment and maintenance

```
1: /* retry_counter and expected_child_num have been initialized to zero when PDC
   starts running */
2: if The current node is not sink and PT is empty then
3:     retry_counter++
4:     if retry_counter > MAX_RETRY_NUM then
5:         retry_counter ← 1
6:         expected_child_num++
7:         if expected_child_num > MAX_CHILD_NUM then
8:             /* Could not find a parent */
9:             Remove grade and schedule
10:            Keep radio on to listen to the channel
11:            retry_counter ← 0
12:            expected_child_num ← 0
13:            return
14:        end if
15:    end if
16:    Generate a PRTS packet
17:    PRTS.child_num ← expected_child_num
18:    Set other fields appropriately and broadcast the PRTS
19: end if
```

one which has the least number of child nodes as its parent[1]. To this end, a variable, *expected_child_num* is defined, such that the node's adjacent lower-grade neighbors at which the number of children is no larger than *expected_child_num* will contend for replying with PCTS after receiving its PRTS. *expected_child_num* is initialized to zero when PDC starts running, and increased by one if it fails in receiving a PCTS, until it reaches *MAX_CHILD_NUM*, the maximum number of children a node with $grade \geq 1$ can have (sink has no such a limit). For each $expected\_child\_num \in [0, MAX\_CHILD\_NUM]$, the node will have at most *MAX_RETRY_NUM* retries to broadcast PRTS. Algorithm 2 shows the above procedure. When *expected_child_num* is increased up to *MAX_CHILD_NUM* but still no parent found, the node will assume that it is in a wrong grade or schedule, and will keep its radio on to find a new grade and schedule to rejoin the network.

On the other hand, after receiving a broadcast PRTS in state `R` from an adjacent upper-grade node, a node needs to determine whether to reply with PCTS to become

[1]There are other metrics that can be considered for a node to determine its parent, such as link quality. With the PRTS/PCTS handshake, it is also easy to implement them in PDC.

Algorithm 3 Determine whether to reply with PCTS after receiving a broadcast PRTS in state `R`

1: **if** PRTS.$grade \neq grade + 1$ **then**
2: /* The PRTS is not from a right grade; do not reply */
3: **return**
4: **end if**
5: **if** The current node is not sink **and** PT is not empty **then**
6: /* A node except sink will not reply its upper-grade nodes (child nodes) if it did not find its parent node */
7: **return**
8: **end if**
9: **if** $table\_length$(CT) >= MAX_CHILD_NUM **or** PRTS.$child\_num$ < $table\_length$(CT) **then**
10: /* Do not reply, to ensure only the nodes with fewer children will reply */
11: **return**
12: **end if**
13: Wait $dur(W)$
14: Generate a PCTS with its fields set appropriately and send it

Algorithm 4 Function $dur(\cdot)$

Input: W
Output: $wait\_duration$
1: **if** The current node is sink **then**
2: $wait\_duration \leftarrow \text{SIFS} + rand(W) \cdot \sigma$
3: **else**
4: $t \leftarrow W/(MAX\_CHILD\_NUM + 1)$
5: $wait\_duration \leftarrow \text{SIFS} + [t \cdot table\_length(\text{CT}) + rand(t)] \cdot \sigma$
6: **end if**
7: **return** $wait\_duration$

the parent of the PRTS sender. The node can be the parent of other nodes if and only if it has found its own parent, otherwise it will not reply. It will not reply, either, if the number of its child nodes reaches MAX_RETRY_NUM or the PRTS sender wants a parent with fewer children, i.e., PRTS.$child\_num < table\_length$(CT). All cases are summarized in Algorithm 3, where $table\_length$(CT) is to obtain the number of items in CT (i.e., the number of child nodes the current node has). If the node can reply, after waiting $dur(W)$, it will unicast a PCTS to the PRTS sender, where $dur(\cdot)$ is a function as shown in Algorithm 4. Since there might be several lower-grade nodes with $grade \geq 1$ receiving the broadcast PRTS and necessary to contend with each other to reply with PCTS, $dur(\cdot)$ ensures that the nodes with fewer child nodes reply

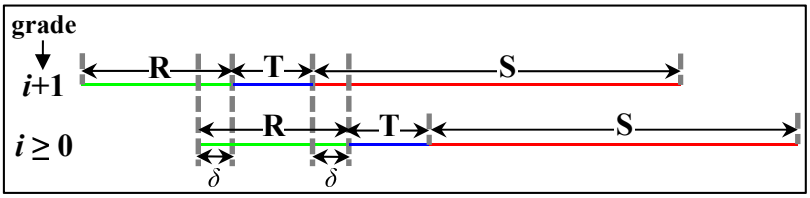


Figure 2.7: Two guard times with the same duration, δ, are added immediately before and after each R state to form a schedule guard region, i.e., $[-\delta, \delta]$.

first.

Once receiving a PCTS reply, the PRTS sender can construct its PT and change to PRTS/PCTS unicast handshake from broadcast. Note that the PCTS replier has not added the sender to its CT, until it receives the first unicast PRTS from the sender. For table update, there are two counters, i.e., *no_pcts_counter* and *no_prts_counter* in PT and CT, respectively. When a counter reaches the pre-determined threshold, the corresponding item in the table will be removed. If the PT of a non-sink node becomes empty, it will change to PRTS/PCTS broadcast handshake to find a parent.

It is possible that a non-sink node receives a unicast PRTS but not intended for itself. If PRTS.*send_num* − *table_length*(CT) > 1, the node has at least two child nodes fewer than the number of child nodes the PRTS sender's current parent node has. To balance the number of child nodes among parent nodes, the node will contend for replying with PCTS (although the received PRTS is not intended) to make the PRTS sender switch to itself as the new parent node. Since the node has fewer child nodes, according to the function $dur(\cdot)$ shown in Algorithm 4, it has a shorter waiting time to send PCTS and thus has a higher probability to win the contention.

2.4.4 Schedule Synchronization

Schedule synchronization is quite significant and challenging to maintain a data-gathering tree established over periodic duty-cycled (sleep-wakeup) radios. To tackle this issue, as shown in Figure 2.7, PDC adds two guard times with the same duration, δ, immediately before and after each R state since the schedule error could be positive or negative as shown in Section 2.3.1. Therefore, a schedule guard region, $[-\delta, \delta]$, for schedule synchronization between nodes is formed. Through the PRTS/PCTS unicast handshake, it is the child node (similar to the *observing* mote shown in Section 2.3.1) which performs synchronization with its parent node (the *reference* mote) when the schedule error (denoted by ε) between them exceeds the guard region, i.e., $\varepsilon \notin [-\delta, \delta]$.

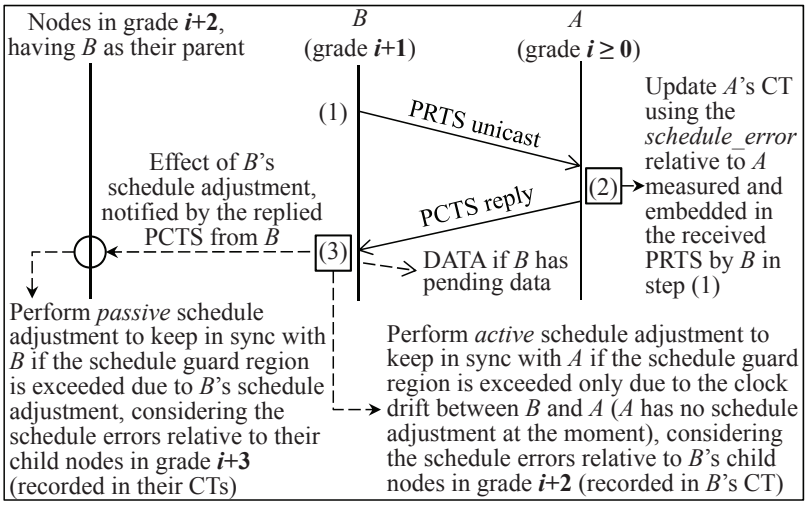


Figure 2.8: PRTS/PCTS unicast handshake between nodes A and B for schedule synchronization and/or data transmission.

Algorithm 5 Step (3) when receiving PCTS

1: Calculate *schedule_error* with the received PCTS
2: Update PT
3: **if** $|schedule\_error| \geq \delta$ **then**
4: $\quad$ *sync_offset* $\leftarrow$ *find_true_offset*(*schedule_error*)
5: $\quad$ Adjust clock using *sync_offset* to be in sync with the parent
6: $\quad$ /* Since the clock has been adjusted, the *schedule_error* stored in PT should be updated accordingly */
7: $\quad$ PT.*schedule_error* $\leftarrow$ *schedule_error* $-$ *sync_offset*
8: **end if**

Note that the node could not use ε as the *offset* directly to adjust its clock to synchronize with its parent, since such an adjustment may affect the synchronization with its child nodes. To solve this issue, after a node receives a designated PRTS from each of its child nodes, the node will record into its CT the schedule error between each child node and itself, which is measured by the child node and embedded in the PRTS. When it is necessary to perform synchronization, the node calculates a true offset to synchronize with its parent node taking into account all these schedule errors stored in its CT, such that it will not lose synchronization with any of its child nodes and thus the pipelined scheduling will not be disrupted.

Taking an example shown in Figure 2.8 for illustration, the details of PRTS/PCTS unicast handshake for the synchronization between a node B and its parent A are shown as below:

Algorithm 6 Function *find_true_offset*$(\cdot)$

Input: *schedule_error*
Output: *sync_offset*

```
1: sync_offset ← schedule_error
2: temp_offset ← schedule_error
3: for Each item ct in CT do
4:     if (schedule_error + ct.schedule_error) ≥ δ then
5:         temp_offset ← δ − ct.schedule_error
6:     else if (schedule_error + ct.schedule_error) ≤ −δ then
7:         temp_offset ← −δ − ct.schedule_error
8:     end if
9:     if |sync_offset| > |temp_offset| then
10:        sync_offset ← temp_offset
11:    end if
12: end for
13: return sync_offset
```

(1) Node B generates a PRTS, and sets the field *schedule_error* to the *schedule_error* calculated in the last handshake and stored in PT, and other fields appropriately, and then PRTS is sent to A. Note that the first *schedule_error* calculated by B is based on a PRTS/PCTS broadcast handshake for tree establishment.

(2) After receiving the PRTS from node B, node A first updates its CT using the embedded information in this PRTS and then replies with a PCTS. As aforementioned, the *schedule_error* obtained from the PRTS and updated in CT will help A to find a true offset when it is necessary to adjust its schedule to synchronize with its parent in grade $i-1$ (if $i \geq 1$), which is similar to the synchronization between B and A, as introduced in step (3) in detail below.

(3) After receiving the PCTS from node A, node B can calculate the *schedule_error* between itself and A, and update the corresponding *schedule_error* field in PT, which will be used in the next handshake to update the CT of B's receiver (i.e., A), as shown in steps (1) and (2) above. With the calculated *schedule_error*, B can determine how to perform schedule adjustment to keep in sync with its parent node A if a synchronization is necessary, as shown in Algorithm 5. Function *find_true_offset*$(\cdot)$ is shown in Algorithm 6, which is to find a true offset for B to perform synchronization with A, taking into account all the schedule errors between B and B's child nodes (as aforementioned, these schedule errors are

stored in B's CT).

There are two kinds of schedule adjustments for synchronization in PDC. A node will perform *active* schedule adjustment if the measured schedule error relative to its parent exceeds the guard region only due to the clock drift between the two nodes. Otherwise, *passive* schedule adjustment will be performed if the schedule error exceeds the guard region due to the schedule adjustment (either active or passive) at its parent. For example, in Figure 2.8, if node A has no schedule adjustment when replying with a PCTS, node B measures the schedule error relative to A after receiving the PCTS and finds that the error reaches or exceeds the guard region, then B will perform the active schedule adjustment. If this adjustment at B makes the schedule error of one of B's child nodes exceed the guard region, the child node will perform the passive schedule adjustment after finding this effect when receiving the PCTS replied by B. For both kinds of schedule adjustments at a node, all the schedule errors between the node and its child nodes should be taken into account to find a proper schedule offset (see Algorithm 5), as mentioned above.

It is worth noting that in Figure 2.8, node B will send DATA packet after receiving the replied PCTS if it has data pending to be transmitted (otherwise, it goes to sleep to save energy). On the other hand, schedule synchronization is meanwhile conducted even if the PRTS/PCTS is for data transmission. In summary, PDC only relies on PRTS/PCTS handshake not only for data transmission as commonly utilized, but also for pipelined scheduling and schedule synchronization over duty-cycled radios, data-gathering tree establishment, and network topology control and maintenance (further discussions are given in the next subsection), all of which are naturally and seamlessly integrated together and able to support each other.

2.4.5 Discussions on Topology Control and Maintenance

Easy Network Deployment. In PDC, when a node is powered on, it listens to the channel without sending any packet, until it receives and parses a packet containing the grade and duty-cycled schedule information to be able to join the network. This is a good feature considering the real case when we deploy a large number of sensor nodes. We do not want the powered-on nodes to start working and sending data traffic before they are deployed to an area, since such actions are useless and may lead to traffic congestion and topology change. Instead, nodes in PDC can only be initialized to work when receiving the traffic initiated by the sink and/or

forwarded by the working nodes which are closer to the sink.

Resilience to Sync Failure. PDC can achieve a good network synchronization performance as shown in Section 2.4.6. However, sync failure may still happen when some unavoidable circumstances happen, such as node rebooting, link breakage when the wireless link quality becomes unacceptable, packet loss due to heavy contention, etc. PDC is robust and resilient to such a failure. Specifically, the *no_pcts_counter* in a node's PT records the number of times of continuously having no PCTS received after broadcasting PRTS. If it reaches a predetermined threshold, the node assumes that a sync failure happens, and then frees its PT by removing the corresponding parent and keeps its radio on to listen to the channel. After parsing the received packets for GDPS (see Algorithm 1), the node can rejoin the network. Compared to the expected life time of a WSN (weeks, months, and in some cases, years), the time used to do the above operations, which varies from seconds to minutes, can be neglected.

Adaptive Response to Topology Changes. PDC is also quite adaptive to topology changes, e.g., some nodes do not work over time, some new nodes are added later, and some nodes move to different locations. Similar to recovering from sync failure, with Algorithm 1 for GDPS, a node can also rejoin the network if topology changes happen, as long as it can receive packets from others closer to the sink.

2.4.6 Implementation and Evaluation

We have built a testbed with PDC implemented in the latest Contiki OS (Contiki 3.0) running on two hardware platforms, namely, Z1 [37] and MicaZ [38] motes as shown in Figure 2.1. Contiki is a pioneer open source OS for the Internet of Things [36]. Nowadays, Contiki can run on over 20 platforms with ten microprocessors supported due to its good portability [103]. In addition, Contiki provides a network simulator called Cooja for the rapid development of sensor networks [104]. One of the significant differences from other network simulators such as OPNET, NS-2/3, and OMNeT++ is that Cooja runs simulations based on the Contiki OS and fully emulated hardware devices. In other words, the code developed for a Cooja simulation can be directly uploaded to real hardware devices even without any change, which makes developing and debugging realistic sensor networks tremendously easier. In this section, we evaluate the synchronization performance of PDC over real Z1 and MicaZ motes, as well as over the fully emulated Z1 mote in the Cooja simulator. To

compare the network performance of PDC versus other protocols, we will only use Cooja with a random network topology, due to its extremely high accuracy when the developed code is uploaded to and running on a real mote, as shown in [105], and for rapid validation and evaluation.

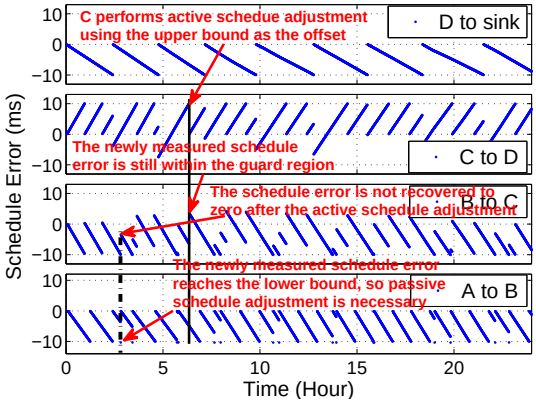
Since there is no hardware clock drift in simulation, we randomly generate clock drift for each node in Cooja according to the schedule errors measured in Section 2.3.1, where a linear clock drift was shown in both real Z1 and MicaZ motes. Note that our protocol is not constrained to linear clock drift. For Cooja simulation, we use two kinds of simulated clock drift to evaluate the synchronization performance of PDC, for both of which, the *clock drift direction* is randomly determined at the beginning and will not be changed in runtime: 1) linear clock drift: there is one clock tick drift[2] every 20 cycles; and 2) nonlinear clock drift: there is one clock tick drift every $rand(20)$ cycles, and therefore, the protocol performance in a more challenging environment can be simulated and evaluated.

Synchronization Performance

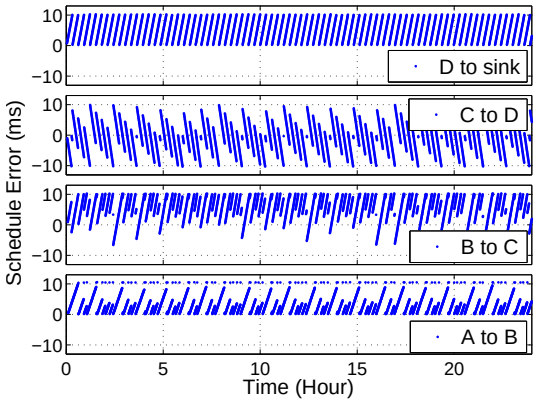
In this subsection, we show the synchronization performance of PDC. The sleep factor ξ in PDC is set to ten and the guard time δ is set to ten ms. A linear topology with five either real Z1, real MicaZ, or fully emulated Z1 motes in Cooja is deployed to form a 4-hop network, denoted as $A \rightarrow B \rightarrow C \rightarrow D \rightarrow sink$. The schedule error between two nodes is measured using the same way as shown in Section 2.3.1. Figure 2.9 shows the schedule errors measured over 24 hours at node A, B, C and D, relative to their parent nodes, i.e., B, C, D and $sink$, respectively. Since the simulation results in Cooja based on the linear clock drift are similar to those from the two real motes, only the simulation results based on the nonlinear clock drift are shown. All results show that the schedule errors can be always within the guard region $[-\delta, \delta]$, which indicates that all nodes can synchronize with each other and there is no disruption on the established pipelined schedules.

According to the synchronization mechanism in PDC, when the schedule error of a node relative to its parent reaches the guard region bound (either the lower bound $-\delta$ or the upper bound δ), it needs to adjust its schedule to re-synchronize with its parent (active schedule adjustment). If it has child nodes, such an adjustment should not affect its synchronization with them. The Function *find_true_offset*$(\cdot)$ in

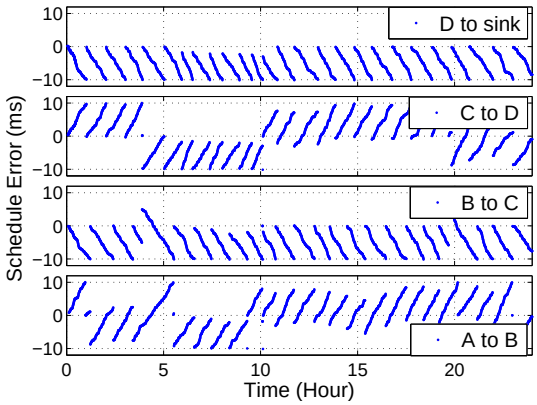
[2]The real-time timer resolution in Contiki for Z1 is $32,768$ Hz, i.e., $32,768$ ticks per second.



(a) Synchronization performance in real Z1 motes



(b) Synchronization performance in real MicaZ motes

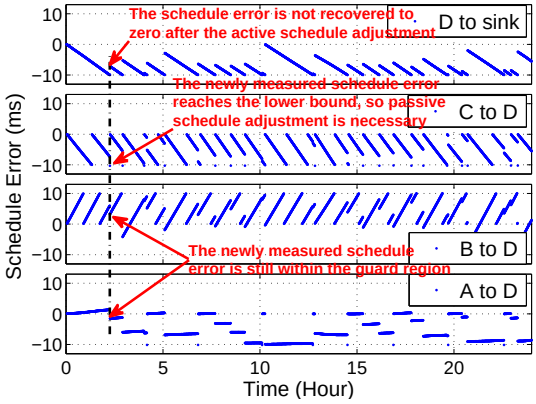


(c) Synchronization performance in fully emulated Z1 motes with nonlinear clock drift

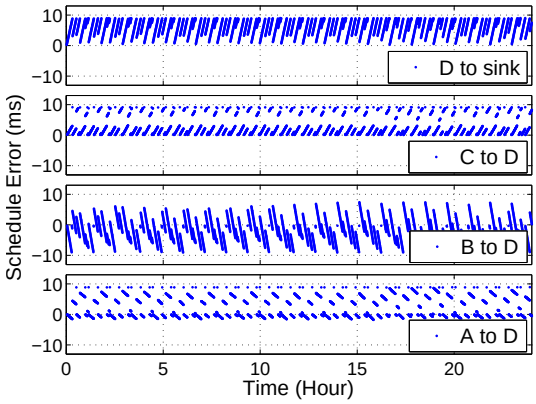
Figure 2.9: Synchronization performance of PDC in a 4-hop linear network: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow sink$, measured over 24 hours.

Algorithm 6 used in Algorithm 5 is for this purpose. If there is no effect on the schedules of its child nodes, the node will use the guard region bound as an offset to adjust its schedule to reset its schedule error to zero. For example, at the time point marked by a solid vertical black line in Figure 2.9a, node C's schedule error relative to D reaches the guard region upper bound, then it will use the upper bound as the offset to adjust its schedule, since the schedule error measured by its child node B relative to the new schedule of C after C's schedule adjustment is changed from negative to positive but still within the guard region. Therefore, B does not need to adjust its schedule, and thus its child node A does not adjust its schedule, either.

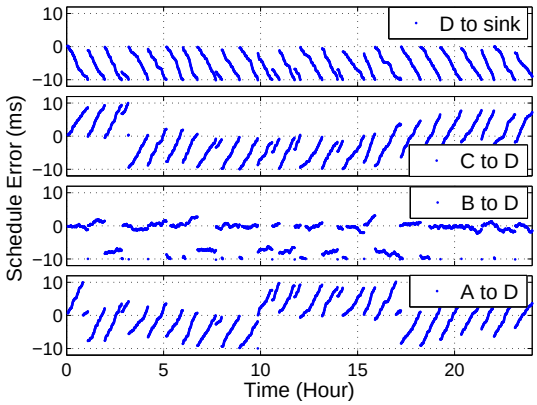
Otherwise, if using the guard region bound as an offset for schedule adjustment will make its child node lose synchronization with itself, the node will not use the bound



(a) Synchronization performance in real Z1 motes



(b) Synchronization performance in real MicaZ motes



(c) Synchronization performance in fully emulated Z1 motes with nonlinear clock drift

Figure 2.10: Synchronization performance of PDC in a 2-hop tree network: $A \rightarrow D$, $B \rightarrow D$, $C \rightarrow D$, $D \rightarrow sink$, measured over 24 hours.

but should determine the schedule offset according to the schedule of its child node. For example, at the time point marked by a dotted vertical black line in Figure 2.9a, if B used the lower bound as the offset to perform active schedule adjustment, then A's newly measured schedule error relative to B would exceed the lower bound of the guard region, and thus A would lose the schedule synchronization with B. To avoid this out-of-sync issue in PDC, B uses the offset calculated based on its CT, which will make the newly measured schedule error of A just reach the lower bound. So B's new schedule error relative to C will not be recovered to zero, and A will perform passive schedule adjustment.

We further investigate the synchronization performance of PDC by changing the linear topology used above to a new topology where node A, B and C all have node D

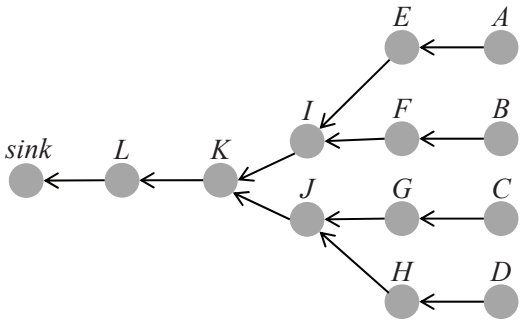
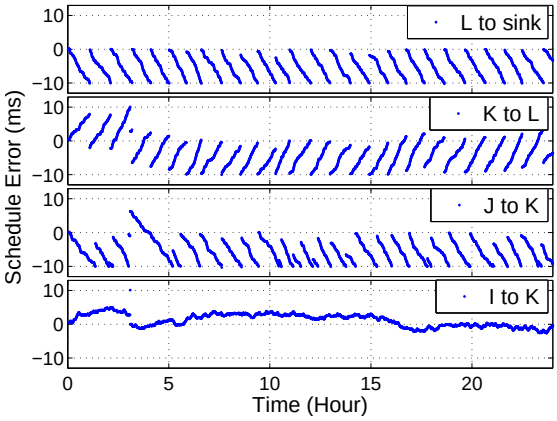


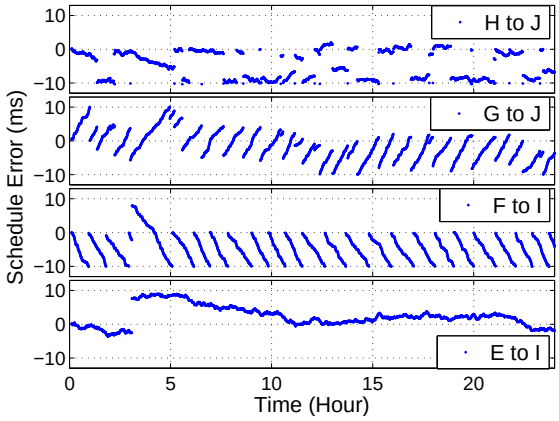
Figure 2.11: A data-gathering tree network for the evaluation of schedule synchronization performance in PDC.

as their common parent node, and D's parent node is the sink. Figure 2.10 shows the schedule errors measured over 24 hours. Also, the established pipelined schedules are maintained quite well. In this case, the schedule adjustment at D needs to consider multiple child nodes and also may not be able to directly use a bound as the schedule offset. Specifically, when the schedule error measured at D reaches the guard region lower/upper bound and thus an active schedule adjustment is necessary, D needs to adjust its schedule by using the bound as the schedule offset if the newly measured schedule errors of all the child nodes relative to D are still within the schedule guard region after this active schedule adjustment at D. Otherwise, according to the function *find_true_offset*$(\cdot)$ in Algorithm 6, the offset is chosen based on the following rule: after D's schedule adjustment, there should be at least one of its child nodes whose newly measured schedule error reaches the guard region bound and none of them whose schedule error exceeds the guard region. Take the time point marked by a dotted vertical black line shown in Figure 2.10a for example. D's schedule error is now reaching the lower bound, so it needs to adjust its schedule error with the offset taking into account all the current schedules of its child nodes (done by *find_true_offset*$(\cdot)$). From Figure 2.10a, we can see that after D's schedule adjustment, its schedule error relative to the sink is not reset to zero, and only C's newly measured schedule error reaches the lower bound but A's and B's are still within the guard region. So only C needs to perform passive schedule adjustment.

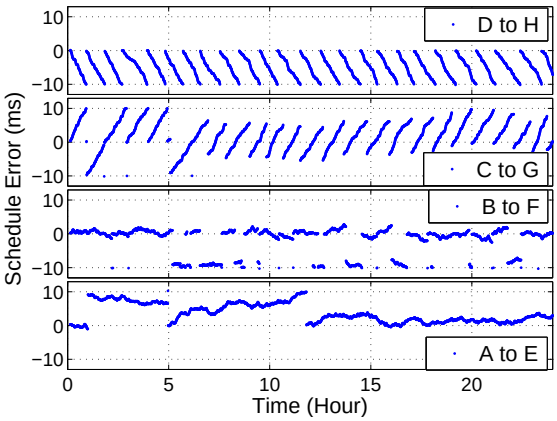
It is worth mentioning that from the Cooja simulation results shown in Figure 2.9c and Figure 2.10c, we can see that the schedule errors measured by node A are in $[-10, 10]$ ms, whereas the errors measured by node A in real Z1 and MicaZ are in either $[-10, 0]$ or $[0, 10]$ ms but not both. This is due to the fact that Cooja uses a randomly generated nonlinear clock drift for each node. It covers the cases which were



(a) Schedule error between L and *sink*, K and L, J and K, and I and K



(b) Schedule error between H and J, G and J, F and I, and E and I



(c) Schedule error between D and H, C and G, B and F, and A and E

Figure 2.12: Synchronization performance of PDC in the network shown in Figure 2.11, based on the fully emulated Z1 in Cooja with nonlinear clock drift.

not reflected in our two real hardware platforms. Therefore, the protocol performance evaluation can be conducted in a more general and challenging environment through Cooja.

Finally, we study the PDC synchronization performance in a tree network a little bit larger as shown in Figure 2.11. To speed up the experiment, and considering that Cooja is based on the fully emulated hardware platforms using the same code as running on the real motes and can be more general to show the protocol performance as shown above, only Cooja simulations are conducted here. Figure 2.12 shows the results. Again, the pipelined schedules established in the network show no disruption caused by out-of-sync schedules.

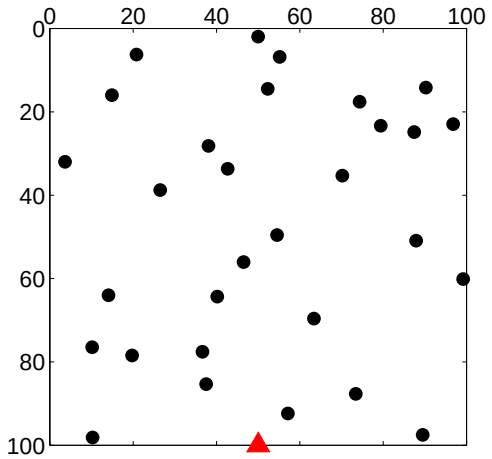


Figure 2.13: A sensor network with 30 nodes randomly deployed in a 100×100 m^2 square area. The sink node shown in red triangle is located in the middle of the bottom bound. The transmission range of each node is set to 30 m, and interference is twice the range of transmission.

Network Performance

In this subsection, the network performance of PDC is evaluated based on the Contiki OS and the fully emulated Z1 mote in Cooja with nonlinear clock drift, in terms of packet delivery ratio, average hop delivery latency, and average duty cycle. The evaluation is based on a sensor network with 30 nodes randomly deployed in a 100×100 m^2 square area, as shown in Figure 2.13. The sink node shown as a red triangle is located in the middle of the lower boundary of the network. The node transmission range in Cooja is set to 30 m and interference is twice the range of transmission.

The performance of PDC is compared with a data collection protocol in the Contiki OS, called Contiki Collect Protocol (CCP), which is designed based on Collection Tree Protocol (CTP) [5], a *de facto* standard for data collection. CTP adopts a data path validation technique to quickly discover and fix routing inconsistencies, and an adaptive beaconing technique to reduce the route repair latency and beacon overhead. CTP contains three components, i.e., 1) routing engine for sending and receiving beacons as well as creating and updating the routing table; 2) forwarding engine for delivering packets between the application layer and MAC layer, detecting and repairing routing inconsistencies (loops) as well as handling duplicate packets; 3) link estimator for one-hop link estimation based on the statistics of ETX (expected transmission count) gathered from beacons and successful transmissions over time.

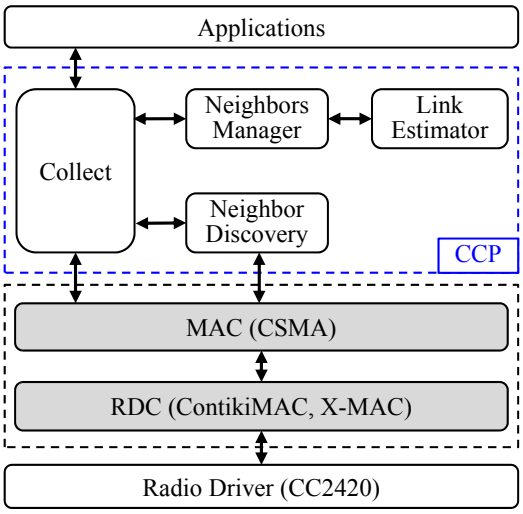
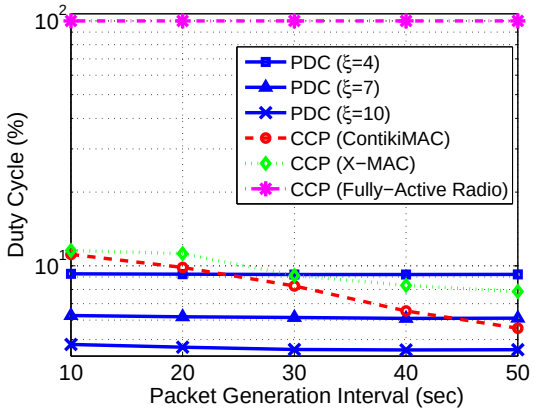


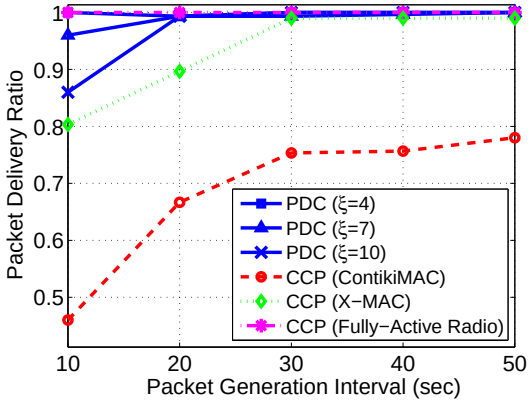
Figure 2.14: The implementation architecture of CCP in the Contiki OS.

CCP is a CTP-like protocol and also the only data collection protocol implemented in the Contiki OS. Figure 2.14 shows the implementation architecture of CCP in the network stack of the Contiki OS, where CCP is located in the network layer of the stack and contains four building blocks, i.e., 1) link estimator, which is similar to CTP's; 2) neighbor manager for recording information about detected neighbors and managing and maintaining routing table; 3) neighbor discovery, for detecting nodes in the communication range of the current node; 4) collect, the central block which provides interfaces for the upper and lower layers, and initializes the other aforementioned blocks in CCP. From Figure 2.14, we can see that the network stack in the Contiki OS is a little bit different from the typical 5-layer model adopted by TCP/IP. In between the network and physical layers, where the MAC layer is usually located, there is an additional layer below MAC, called Radio Duty Cycling (RDC), which is responsible for switching radio between on and off, whereas the MAC layer adopting a CSMA mechanism is responsible for avoiding collisions at the radio medium and packet retransmission if there is a collision. In contrast to CCP, PDC integrates routing and MAC naturally, as shown in Figure 2.4.

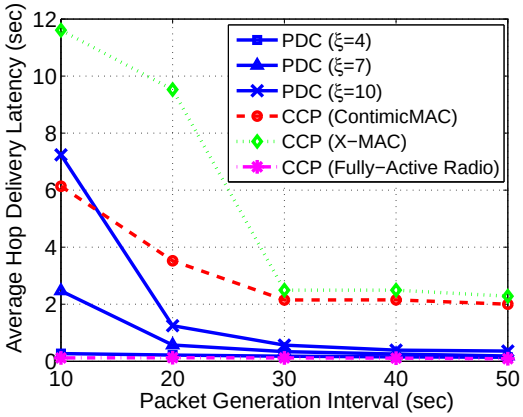
The latest Contiki OS provides two RDC mechanisms, i.e., ContikiMAC (the RDC mechanism provided by the Contiki OS) and X-MAC, both of which will be used underlying CCP in our comparison (in addition, the CCP with no RDC mechanism, i.e., radio is fully active always, is also compared). X-MAC [106] is one of the most typical and widely applied MAC protocols. It is a duty-cycled MAC based on short preambles initiated by a sender to achieve rendezvous with its receiver. It has been



(a) Average Duty Cycle (y-axis is in log scale)



(b) Packet Delivery Ratio



(c) Average Hop Delivery Latency

Figure 2.15: Network performance of PDC in terms of packet delivery ratio, average hop delivery latency, and average duty cycle.

enhanced in the Contiki OS with a technique called phase lock, which is also adopted in ContikiMAC. With this technique, a sender can learn when the intended receiver will turn its radio on to listen to the channel. ContikiMAC also adopts a mechanism called fast sleep optimization, which lets potential receivers go to sleep earlier to save energy if CCA (Clear Channel Assessment) is triggered due to radio noise.

For PDC, the sleep factor ξ is set to seven, so that PDC and X-MAC can roughly have the same duty cycle (around 6.35%) when the network is idle (i.e., no data packet in the network). On the other hand, the wakeup frequency in ContikiMAC is set to 32 Hz. Each wakeup takes about 0.6 ms to perform CCA, leading to a very low duty cycle (around $(32 \times 0.6/1,000)/1 = 1.92\%$) when the network is idle. The PDCs with a smaller ξ ($= 4$) and a larger ξ ($= 10$) are also compared. All 30 nodes in

the network can generate data packets. The Packet Generation Interval (PGI) varies from ten to 50 sec with a step of ten sec.

Figure 2.15 shows the comparison results in terms of average duty cycle, packet delivery ratio, and average hop delivery latency. Since PDC adopts a fixed duty-cycled pattern (determined by ξ), from Figure 2.15a we can observe that the duty cycle in PDC decreases very slightly as PGI increases. In contrast, the duty cycles in ContikiMAC and X-MAC are more dynamic, since they can switch radio between on and off according to the data traffic amount. However, they do not comprehensively consider the coordination with the CCP in the network layer, which may lead to a poor network performance. By observing the results of the three performance metrics shown in Figure 2.15, we can find that PDC has the most energy-efficient performance. With the lowest $\xi = 4$ corresponding to the highest duty cycle, the average duty cycle in PDC over the five PGIs (10–50 sec) is close to that in the CCP with either ContikiMAC or X-MAC. Meanwhile, PDC with $\xi = 4$ has 100% packet delivery ratio, and a delivery latency performance much close to the CCP with no RDC mechanism (i.e., all nodes are working in the fully-active mode). Also, all nodes in the network are synchronized with each other quite well.

2.5 Adaptive Data Collection

Based on PDC, ADC involves a free-addressing scheme during the topology establishment and adopts a dynamic duty-cycling scheme for data transmission, as shown in Sections 2.5.1 and 2.5.2, respectively. Then Section 2.5.3 shows the implementation and evaluation of ADC.

2.5.1 Topology Establishment with Free Addressing

Once a node has identified its grade and schedule, it will leverage the periodic PRTS/PCTS broadcast handshake to establish a data-gathering tree for data collection, as introduced in Section 2.4.3. To avoid the cost and overhead of assigning a unique address to each sensor node during the manufacturing phase or performing an independent address allocation, exchange, and management mechanism in run-time, a node in ADC will use an RID for packet communication. Specifically, a node broadcasts a PRTS embedding its RID. There might be several adjacent lower-grade nodes receiving the PRTS and thus necessary to contend with each other for replying

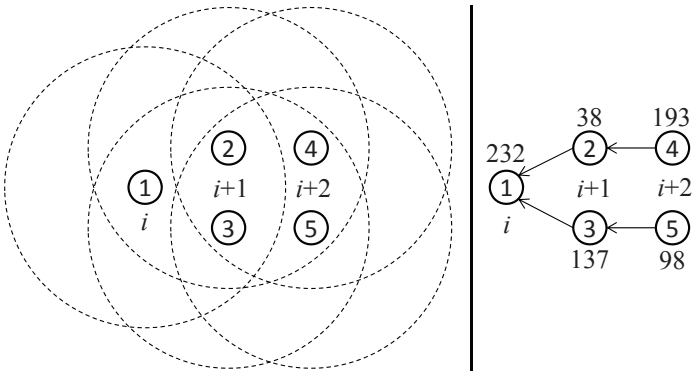


Figure 2.16: Illustration of free addressing. For the left part of the vertical line, each dashed circle shows the node's communication range and the number inside each node shown as a solid circle is the node name, while for the right part, the number beside each node is its RID and the arrows indicate the network topology. Node 1 is in grade i ($i \in \mathbb{N}^0$), nodes 2 and 3 are in grade $i+1$, and nodes 4 and 5 are in grade $i+2$.

with PCTSs with their RIDs embedded as well. The lower-grade node which successfully wins the contention will add the PRTS sender into its CT, and the PRTS sender will add the PCTS sender into its PT after receiving the replied PCTS. Then the two nodes will change from the broadcast handshake to unicast handshake for data transmission, as well as for schedule synchronization and topology control and maintenance. Figure 2.16 shows an example of the above procedures for illustration, where node 1 is in grade i ($i \in \mathbb{N}^0$), nodes 2 and 3 are in grade $i+1$, and nodes 4 and 5 are in grade $i+2$.

Since RID is randomly generated, RID conflict is inevitable. It does not matter if two nodes in different grades have an identical RID, since their grades can be utilized to differentiate them. It may also happen that several nodes in the same grade and having the same parent have generated an identical RID, which will cause an RID conflict issue. Taking the communications among nodes 1, 2, and 3 in Figure 2.16 for example, the following scheme is adopted in ADC to effectively solve the RID conflict between nodes 2 and 3. Assuming one of them fails in the PRTS broadcast contention, say node 3, then

- if node 3 can overhear the PRTS from node 2, it will notice the conflict and thus regenerate a new RID; otherwise,
- if node 3 can receive the PCTS replied to node 2 from node 1, node 3 can also notice the conflict and regenerate a new RID; otherwise,

- node 3 could not notice the conflict. Assume node 3 wins the contention and broadcasts an PRTS successfully next time. Every time a node receives a *broadcast* PRTS (in contrast to *unicast* PRTS), it will look up its CT to find whether there is an RID identical to the RID of the PRTS sender. So node 1 will notice the conflict after receiving the PRTS from node 3. It will generate a different RID and embed it in its replied PCTS for node 3.

Note that a node in ADC will not reply with PCTS after it receives a broadcast PRTS from any higher-grade node, until it determines its RID and parent in the lower grade. For example, node 3 will not be set as a parent of node 4 or 5 if it has not yet determined its RID and parent (node 1). In addition, it would be also possible that node 1 could successfully receive both the broadcast PRTSs from nodes 2 and 3 if they could not hear each other during their contention (i.e., they are hidden terminals to each other). Since node 1 can notice whether there is an RID conflict or not between nodes 2 and 3, it will set a flag in its replied PCTS to notify nodes 2 and 3 of regenerating their RIDs, if they have identical RIDs.

ADC only needs to handle the RID conflict among those nodes which have the same parent. Because, even if two nodes in the same grade have an identical RID, their data transmissions will not be affected as long as they have a different parent. Therefore, ADC can use a small integer range for uniformly at random generating an RID for each node. In the ADC implementation in the Contiki OS, we use an 8-bit variable for RID with a range of $[0, 255]$ (as shown in Figure 2.16), since the number of the children a node has is usually much less than 256.

2.5.2 Data Transmission with Dynamic Duty-Cycling

The number of slots in state `S`, i.e., sleep factor ξ, determines how long a radio can sleep during a cycle. The more slots state `S` contains, the lower duty cycle a node radio has and correspondingly the more energy can be saved. However, a low duty cycle will lead to a long packet delivery latency and low network throughput, especially under heavy traffic loads, while a high duty cycle will lead to an unwanted energy consumption, especially under light traffic loads. Therefore, a dynamic duty cycle adaptive to the network traffic load is highly expected in a protocol design for the data collection of a sensor network. ADC can adaptively utilize the sleeping slots in state `S` to achieve a Dynamic Duty-Cycling (DDC).

To this end, the PRTS packet contains a DDC flag, which will be set if a node

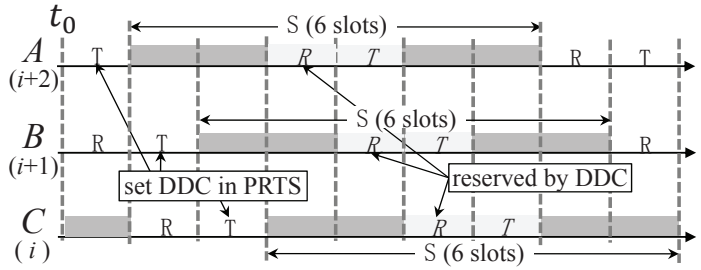
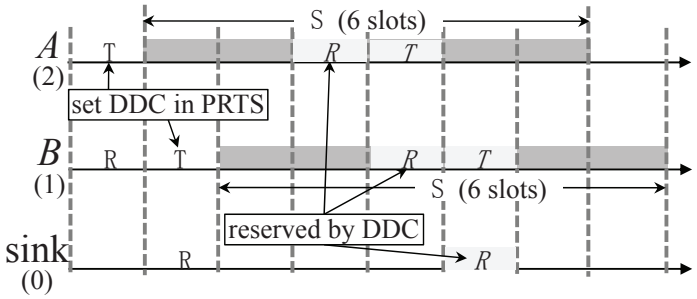


Figure 2.17: Non-last-hop transmission along a path $A \to B \to C$, where node A, B, and C are in grades $i+2$, $i+1$, and i ($i \in \mathbb{Z}$ and $i \geq 1$), respectively.

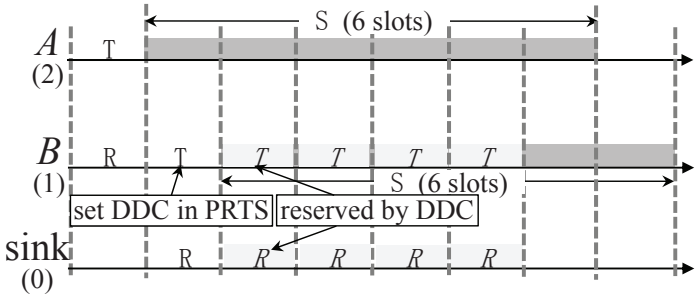
has more than one packet to be transmitted, so that its receiver would wake up in state `S` instead of waiting for the next state `R` to receive the left packet. Depending on whether it is a last-hop transmission or not, the dynamic duty-cycling scheme in ADC will be discussed under following two cases.

Non-Last-Hop Transmission

Figure 2.17 shows an example for illustration about the data transmissions along a path $A \to B \to C$, where node A, B, and C are in grades $i+2$, $i+1$, and i ($i \in \mathbb{Z}$ and $i \geq 1$), respectively. Assume node A receives a PRTS with the DDC flag set from a child node in grade $i+3$ in the `R` state before t_0 (the part before t_0 is not shown in the figure). Then node A will adjust its schedule to reserve two continuous sleeping slots in state `S` (`R` and `T`, shown in italic in the figure), which are used for receiving one more data packet from that child and forwarding the received data packet to its parent (node B), respectively. Meanwhile, in the upcoming `T` state after t_0, node A will also set the DDC flag in its own PRTS. So do nodes B and C, and thus the data packet can still be forwarded in a pipelined fashion in state `S`. Considering interference is about twice the range of transmission for the wireless communication in an open-space environment [100], after any pair of states `R` and `T`, a node has to sleep at least two slots before waking up again for another data transmission, so that it will not interfere with its previous/next-hop neighbors. Therefore, the sleeping slots in state `S` cannot be utilized for data transmissions unless `S` contains at least six slots, and the maximum number of sleeping slots in state `S` that a node can utilize for data transmissions is $\left\lfloor \frac{\xi-2}{4} \right\rfloor \times 2$.



(a) Node A reserves the sleeping slots in state `S` for data transmission.



(b) Node A has no data packet to transmit in state `S`.

Figure 2.18: Last-hop transmission along a path $A \rightarrow B \rightarrow$ sink. There are two cases: (a) node A reserves the sleeping slots in state `S` for data transmission, and (b) node A has no data packet to transmit in the state `S`.

Last-Hop Transmission

The data transmissions along a path $A \rightarrow B \rightarrow$ sink shown as in Figure 2.18 are utilized for illustration. There are two cases for the last-hop transmission, depending on whether or not a grade-two node (e.g., node A in Figure 2.18) either receives a PRTS with the DDC flag set in state `R` or sets the DDC flag of its own PRTS in state `T`. If so, it is similar to the non-last-hop transmission, as shown in Figure 2.18a, except that it is not necessary for node B to set the DDC flag of its PRTS packet to inform the sink node of waking up, since the sink node could be always awake for collecting data. Otherwise, if node A does not reserve its sleeping slots for data transmission, all the sleeping slots of node B except the two immediately before the next regular `R` state can be utilized for transmitting data, as shown in Figure 2.18b, since they will not interfere with the previous-hop transmissions.

Based on the above schemes, ADC can adaptively adjust its duty cycle to the network traffic load by utilizing the sleeping slots on demand. If the network traffic load is light, the regular `R` and `T` slots are enough for data delivery, while with a heavy

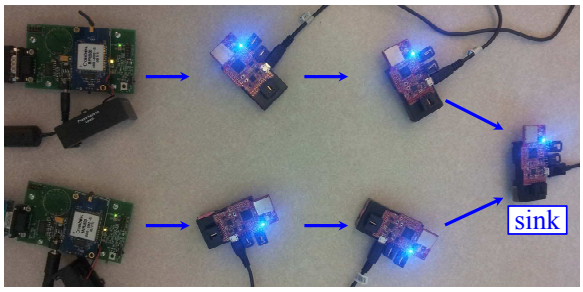


Figure 2.19: A testbed consisting of five Z1 and two MicaZ motes forming a 3-hop network for protocol performance evaluation.

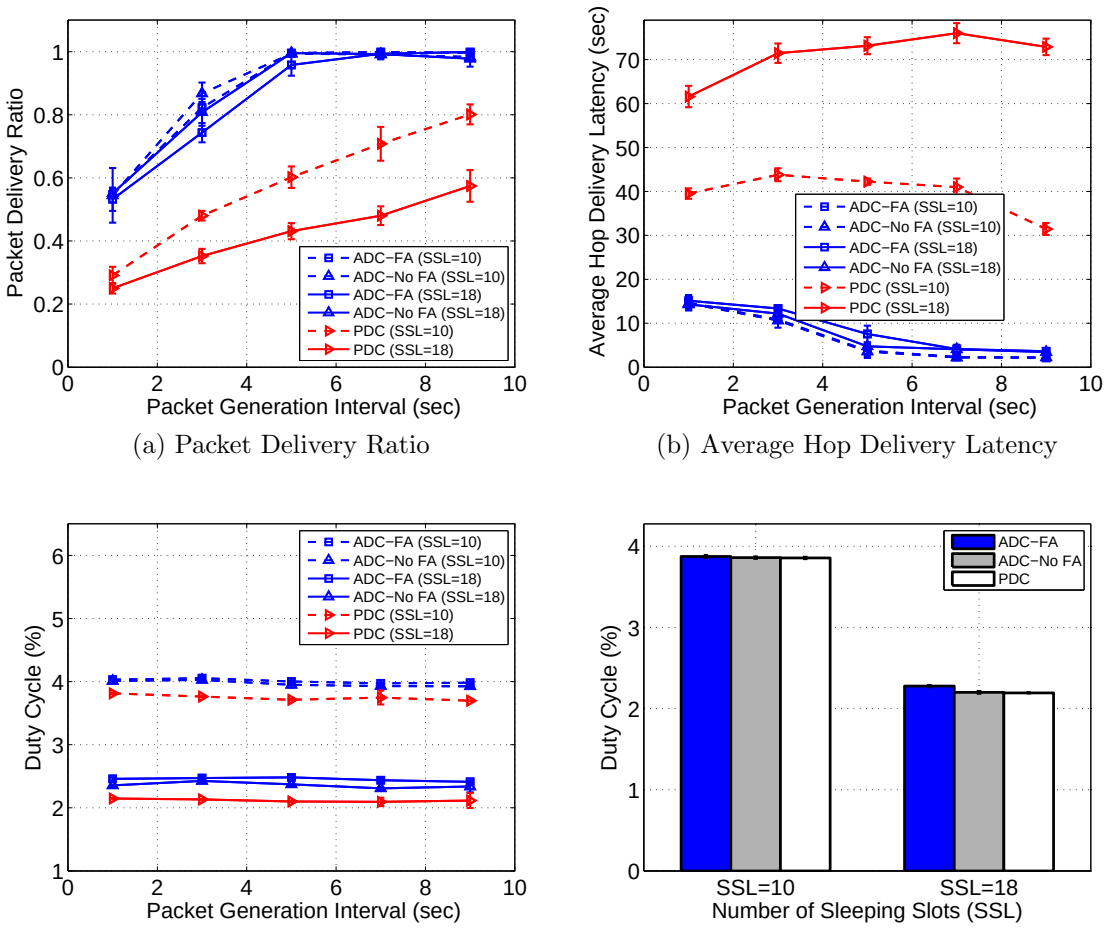
traffic load, the sleeping slots will be utilized efficiently.

2.5.3 Implementation and Evaluation

ADC has also been implemented in the Contiki OS and evaluated in a testbed consisting of five Z1 and two MicaZ motes. In this subsection, we evaluate the network performance of ADC in both the testbed and a larger Cooja simulation consisting of three platforms using the same code, in terms of packet delivery ratio, average hop delivery latency, and average node duty cycle. PDC is used as a benchmark for comparison. We have shown in Section 2.4.6 that PDC outperforms the *de facto* standard, i.e., CCP, in Contiki OS for data collection. Therefore, for ease of presentation, the comparison with CCP is not shown in this subsection.

Performance Evaluation in Testbed

In the testbed, there are six source nodes and one sink node, forming a 3-hop network with the topology shown as in Figure 2.1. Each source node will generate 50 packets, with PGI varying from one to nine seconds. At most five retransmissions are allowed for each packet before dropping it. The queue buffer size is set to ten packets. A unique ID is preassigned to each node. Both the ADCs with and without Free Addressing (FA) are evaluated, denoted as "ADC-FA" and "ADC-No FA", respectively. In ADC-FA, nodes do not use the preassigned IDs but the RIDs generated in runtime. The number of Sleeping SLots (SSL) contained in `S` is set to ten or 18, and thus for ADC, the maximum number of times that a node can wake up during its state `S` for data transmission is two or four, respectively (i.e., at most $2 \times 2 = 4$ or $4 \times 2 = 8$ sleeping slots can be utilized for data transmission, respectively). We run each experiment three times for a duration of 40 minutes.



(a) Packet Delivery Ratio

(b) Average Hop Delivery Latency

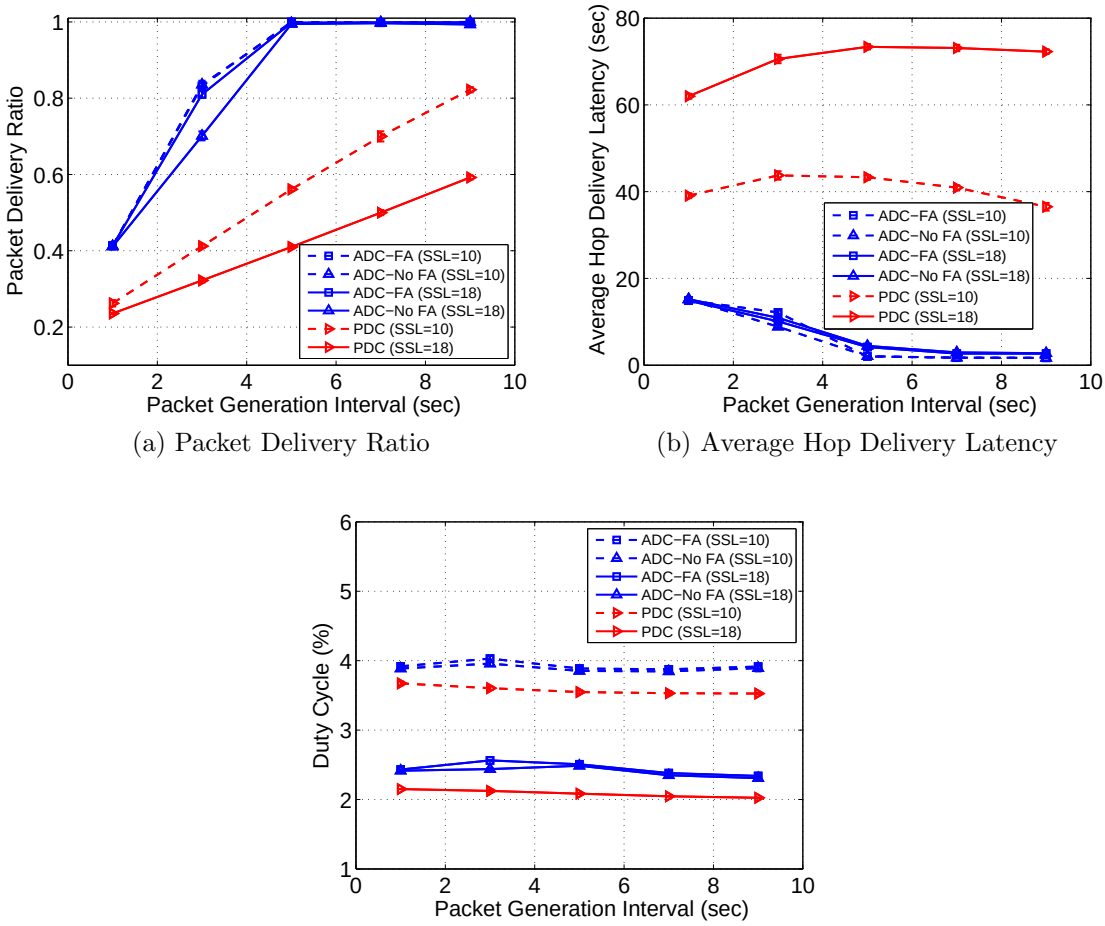
(c) Average Duty Cycle

(d) Average duty cycle when the network is idle

Figure 2.20: ADC performance evaluation based on the testbed shown in Figure 2.1.

Since PDC cannot utilize the sleeping slots for data transmission, the packet delivery ratio in PDC is much lower than in ADC, as shown in Figure 2.20a. For PDC, a larger SSL will lead to a lower packet delivery ratio, because if a packet fails in being transmitted in the current cycle, it has to wait SSL slots for the next cycle, and thus the node queue becomes full faster, resulting in more packet loss. In contrast, SSL affects ADC very slightly.

SSL also has a very slight effect on the average hop delivery latency in ADC, as shown in Figure 2.20b. However, with a larger SSL in PDC, a packet has a longer waiting time on average, leading to a larger delivery latency. In addition, as PGI increases, the packet delivery latency in PDC increases first and then decreases. Because as PGI increases, more packets can reach the sink node but with longer queue



(a) Packet Delivery Ratio

(b) Average Hop Delivery Latency

(c) Average Duty Cycle

Figure 2.21: ADC performance evaluation in the Cooja simulations based on a network the same as in the testbed shown in Figure 2.1.

waiting time, which contributes to increasing the latency as a dominating part, while as PGI continuously increases, packets can be forwarded faster, which dominates and decreases the latency. In contrast, the packet delivery latency in ADC decreases always, and finally has almost no change when the queue waiting time of a packet cannot be decreased any more.

As shown in Figure 2.20c, with a larger SSL in PDC, a node can have more time to be in state `S`, corresponding to a lower duty cycle. ADC has a similar performance, because the least number of sleeping slots where a node in the ADC with SSL=18 has to be asleep is larger than in the ADC with SSL=10 ($18 - \lfloor \frac{18-2}{4} \rfloor \times 2 = 10$ *vs.* $10 - \lfloor \frac{10-2}{4} \rfloor \times 2 = 6$). In addition, the duty cycle in both ADC and PDC has almost no change as PGI increases, due to the fact that the number of packets to be transmitted

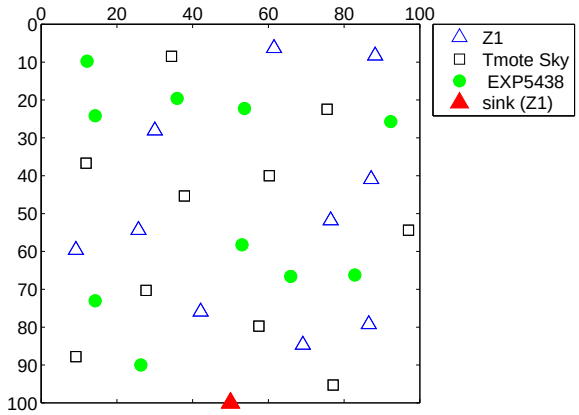


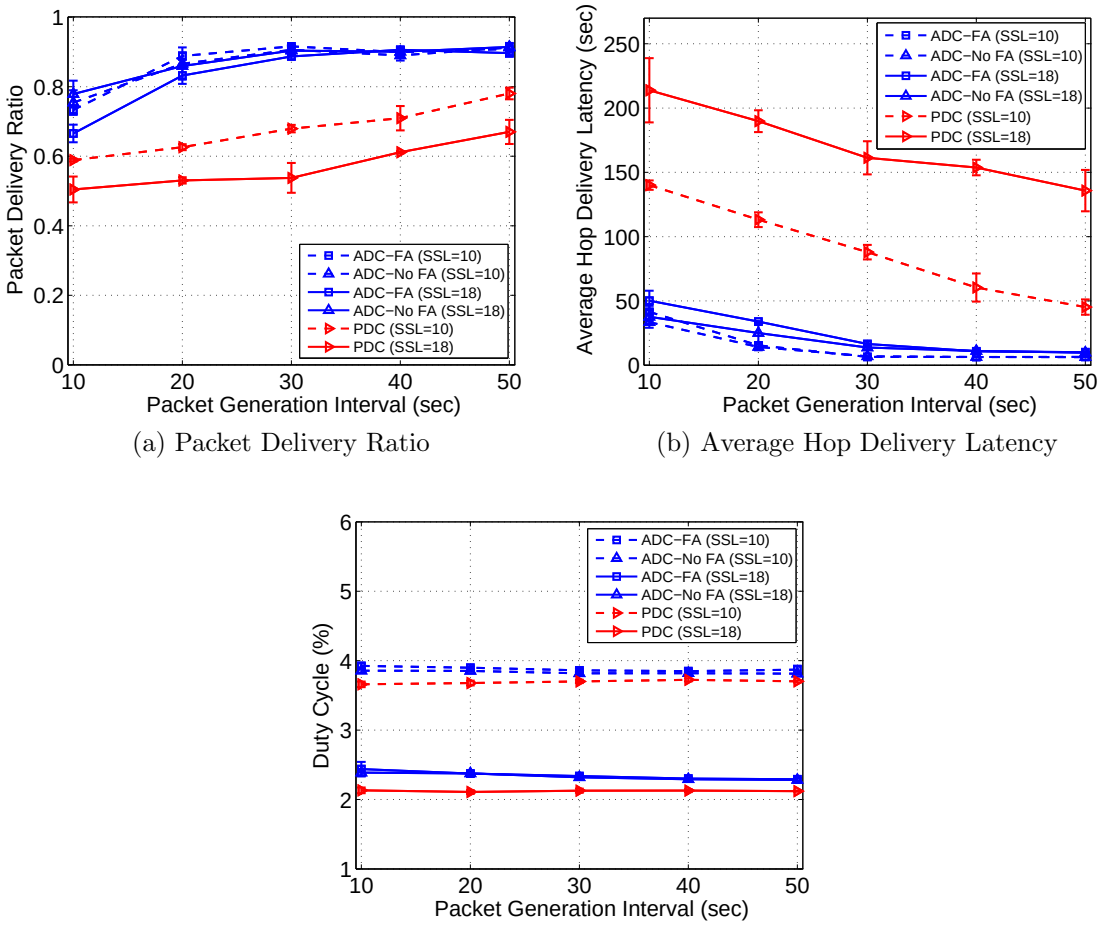
Figure 2.22: A sensor network consisting of three platforms (i.e., Z1, Tmote Sky, and EXP5438, each has ten motes), and randomly deployed in a 100×100 m^2 square area, with the sink node located in the middle of the lower boundary. The node transmission range is set to 30 m, and the interference range is 60 m.

is fixed. Since ADC can use the sleeping slots for data transmission, the duty cycle in ADC is a little higher than in PDC. As the network becomes idle, ADC adaptively does not wake up in state `S` and increasingly achieves the same duty cycle as PDC, as shown in Figure 2.20d.

For setting an appropriate SSL, obviously, there is a tradeoff in PDC among the three metrics. For example, a larger SSL can conserve more energy but lead to a lower packet delivery ratio and longer latency. On the contrary, a larger SSL is expected in ADC for conserving more energy, without obviously affecting the other two performance metrics. Furthermore, all the results in Figure 2.20 show that the ADCs with and without free addressing have almost the same performance, which demonstrates the efficacy of the proposed free-addressing scheme.

Performance Evaluation in Cooja

Using a network the same as the testbed. We have also conducted Cooja simulations with five fully emulated Z1 and two fully emulated Tmote Sky [40] (currently, Cooja does not support the communication between the emulated MicaZ and non-MicaZ platforms) based on the same network as in the testbed, using the same code uploaded to the real Z1 and MicaZ. Similarly, we run each simulation three times for a duration of 40 minutes. The evaluation results are shown in Figure 2.21.



(a) Packet Delivery Ratio

(b) Average Hop Delivery Latency

(c) Average Duty Cycle

Figure 2.23: ADC performance evaluation in the Cooja simulation based on the random network shown in Figure 2.22.

Although the simulation results are not exactly the same as those obtained from the testbed (due to the involved software randomness, nonlinear simulated clock drift, etc.), the two sets of results have almost the same range and trend, which shows that the Cooja simulator can be safely and accurately utilized for protocol design and validation and performance evaluation (as also shown in the prior work [105]). To accelerate the evaluation progress, we use a larger network in Cooja simulations below to evaluate the ADC performance.

Using a larger network. Figure 2.22 shows a larger network in Cooja simulator, which consists of three platforms, i.e., Z1, Tmote Sky, and EXP5438 [41] (each has ten motes) deployed randomly at uniform in a 100×100 m^2 square area, with the sink node (Z1) shown as a red triangle located in the middle of the lower boundary. Cooja

assigns an ID to each node as its MAC address in a sequential way starting from one. Likewise, these IDs will not be utilized by ADC-FA. The transmission range of each node is set to 30 m and interference is twice the range of transmission. Each source node will generate 20 packets, with the PGI varying from ten to 50 seconds. The retransmission limit of each packet is set to ten. The queue buffer size of each node is set to 15 packets. We run each simulation three times for a duration of 60 minutes.

Figure 2.23 shows the obtained evaluation results. Similar to the results obtained from the testbed, the simulation results show that ADC outperforms PDC considering all the three performance metrics comprehensively. With a different SSL, there is an obvious tradeoff in PDC among the three metrics, while ADC prefers to select a relatively large SSL to save more energy without obviously affecting the performance on packet delivery ratio and latency. Also, the proposed free-addressing scheme has almost no effect on the ADC performance.

2.6 Conclusions

This chapter presented a pipelined data collection protocol, i.e., PDC, for duty-cycled sensor networks, to comprehensively achieve high energy efficiency, low packet delivery latency, and practical and effective schedule synchronization. By only relying on PRTS/PCTS handshake with a cross-layer integration design, all the components in PDC are naturally and seamlessly integrated together and able to support each other. PDC has been implemented on the latest Contiki OS. The practicality and efficacy of PDC has been extensively demonstrated. We first validated the synchronization performance of PDC based on two real hardware platforms as well as the fully emulated motes in Cooja with nonlinear clock drift (which provides a more general and challenging evaluation environment). Then we evaluated the network performance in comparison with the Contiki Collect protocol with different underlying RDC mechanisms provided by Contiki.

Based on PDC, an adaptive data collection protocol, ADC, was also presented. With the dynamic duty-cycling feature, ADC can utilize the sleeping slots adaptively for data transmission, improving the network load adaptivity and energy efficiency. With the free-addressing feature, there is no need to pre-assign unique addresses to sensor nodes or perform any address allocation and management procedure in runtime, improving the network heterogeneity. The implementation of PDC and ADC in the Contiki OS can be found in [107].

Chapter 3

Protocol Modeling and Analysis

3.1 Overview

In this chapter, we propose a framework to model a class of duty-cycled data collection protocols for Linear Sensor Networks (LSNs), such as PDC proposed in Chapter 2. After the related work, we first analyze the MAC process in the protocol, and then the queueing behavior with and without retransmission. By combining the two parts of the model, we can analyze the network performance in terms of the throughput, energy efficiency, and packet delivery latency. Finally, the model is validated through extensive OPNET simulations.

3.2 Related Work

The research on LSNs has recently attracted increasing attention due to the vast requirements on the monitoring and surveillance of a structure or area with a linear topology, where the linearity is considered to increase the communication quality and efficiency in routing, network reliability, robustness, fault tolerance, energy efficiency, and network lifetime [9]. There are various promising applications of LSNs, such as pipelines, railroad/subway/bridge, highway traffic, and international border monitoring. Existing research works on LSNs involve the framework and architecture design [10, 11, 108], protocol and algorithm design (including routing protocol [109–111], node placement and network coverage [112–115], MAC protocol and scheduling [12, 116–119], etc.), and network modeling and analysis [112, 114, 120–123].

Specifically, for the LSNs modeling and analysis, Phelan et al. [112] studied the

problem of maximizing the wireless connection lifetime between a stationary transmitter and receiver using mobile relays that are initially placed arbitrarily on the line between the transmitter and the receiver. In [114], an analytical model is developed to estimate the impact of the node faults on the network coverage in LSNs. MomČilović et al. [120] derived the maximum throughput of the network and established the packet arrival process that maximizes the throughput with the proposal of a stochastic linear network model. Noori et al. [121] analyzed the traffic load distribution over the nodes randomly deployed in an LSN with the shortest-path routing. With the analysis, the location of the nodes forming the network bottleneck is identified. In [122], an analytical model with reliability analysis for LSNs is developed. Mohamed et al. [123] developed an analytical model to estimate the sensing coverage in the presence of node faults. Overall, however, there is few work on the modeling and performance analysis for LSNs based on a MAC protocol adopting the duty-cycling mechanism, which is utilized to alleviate the energy efficiency problem—one of the most crucial issues in WSNs.

On the other hand, for a WSN without limiting the network topology to a linear one, there has been some work done for modeling a duty-cycled MAC. Zhang et al. [124] proposed a model to evaluate the performance of S-MAC [125] under different traffic and channel conditions. Yang et al. proposed two Markov models in [126] to analyze the throughput of S-MAC with and without retransmissions, respectively. A similar approach is used in [127] for the throughput analysis of X-MAC [128]. The work was then extended in [129] to analyze the throughput, delay, and energy consumption for both S-MAC and X-MAC. Donmez et al. [130] also derived the expected throughput and energy consumption for S-MAC, however, under the saturated load condition. Guntupalli et al. [131] proposed a model to evaluate the performance of S-MAC for the aggregated packet transmission mode. Lee et al. [132] proposed a model for RIX-MAC [133] in a fully-connected network without supporting retransmission. Different from the proposed model in this chapter for a multi-hop LSN, all the above models were applied to a single-hop scenario. In [134] and [135], Luo et al. proposed a continuous-time Markov chain model to analyze the performance of a contention-based MAC protocol. A similar approach was adopted to analyze the performance of S-MAC in [136]. Although the model is for a multi-hop scenario, it is only applicable to the saturated load condition. Due to this limitation, the number of contending neighbors of a node is just the number of neighbors within the sensing range of this node, but this in fact should be determined by the stationary probabil-

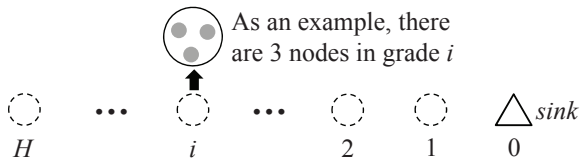


Figure 3.1: LSNs model (the numbers below the triangle (sink) or dashed circles represent the grade; each dashed circle contains one or more nodes in the same grade).

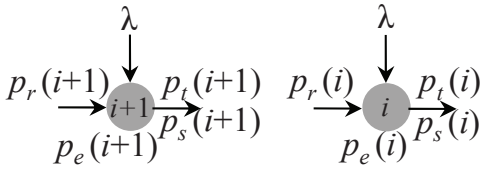


Figure 3.2: There are two nodes respectively in grade i ($i \geq 1$) and $i+1$, where p_t: the probability of winning the contention for channel access; p_s: the probability of successfully transmitting a packet; p_r: the probability of successfully receiving a packet; p_e: the stationary probability that the node transmission queue is empty; λ: the packet generation rate.

ity of the empty-queue state of each node. The model proposed for the performance analysis of DMAC [137] in [138] is only applicable to a network with single source node, while in our model, all nodes in the network can operate as source nodes.

3.3 Modeling and Analysis

3.3.1 System Model and Assumptions

As shown in Figure 3.1, all nodes in the linear topology are divided into H grades (shown as the dashed circles) according to their minimum hop distances to the sink node. For example, the sink node is marked in grade zero, its immediate neighbors are marked in grade one, all these nodes' unmarked immediate neighbors are marked in grade two, and so on. Therefore, each dashed circle contains one or more nodes. All nodes in the network generate data independently following a Poisson process (which has been justified in [29]) with a rate of λ packets/sec. Except the grade-1 nodes which send packets directly to the sink, each node has one node in the adjacent lower grade to relay its packets. The capacity of the FIFO queue in each node is set to K packets. The channel is assumed to be ideal without channel fading and capture effect as in [129].

We choose PDC with the pipelined-scheduling feature proposed in Chapter 2 as the underlaying MAC for a multi-hop LSN. The proposed analytical model contains two parts, namely, the analysis on the MAC process of PDC (by modifying this part, we can easily generalize the model to other duty-cycling MACs with the same pipelined scheduling feature) and the Markov-chain queueing model with and without retransmission. With the combination of these two parts, we obtain the probabilities related to the packet transmission at a node (such as the probability of winning the channel and that of transmitting successfully) and the stationary queue length distribution. These results are further utilized to analyze the average packet delivery latency, average active time ratio per cycle per node, and system throughput for the network model in Section 3.3.1. Figure 3.2 summarizes the probabilities involved in our model.

3.3.2 Analysis on the Protocol Process

Let $p_t(i)$ denote the probability that a node in grade i can transmit, i.e., it wins the contention for accessing the channel, $p_s(i)$ the transmission is successful, $p_c(i)$ the transmission fails due to collision, and $p_b(i)$ the channel is sensed busy. For a node in grade $i-1$ ($i \geq 1$), suppose there are N_i nodes in grade i within its communication range. If one of them as an observed node has a packet to send, the probability that n_i ($n_i \in [0, N_i - 1]$) out of the other $N_i - 1$ nodes are also contending for the channel is

$$\begin{aligned} C_{n_i} =& \binom{N_i - 1}{n_i} \cdot (1 - p_e(i))^{n_i} \cdot p_e(i)^{N_i - 1 - n_i} \\ =& \frac{(N_i - 1)!}{n_i!(N_i - 1 - n_i)!} \cdot (1 - p_e(i))^{n_i} \cdot p_e(i)^{N_i - 1 - n_i}, \end{aligned}$$

where $p_e(i)$ is the stationary probability that the transmission queue of a node in grade i is empty when it enters its `T` period in every cycle. Given that n_i ($n_i \in [0, N_i - 1]$) nodes are contending, the probability that the observed node wins the contention (no other nodes in the same grade pick a smaller backoff counter) is

$$t_{n_i} = \sum_{w=0}^{W-1} \frac{1}{W} \left(\frac{W - w}{W} \right)^{n_i},$$

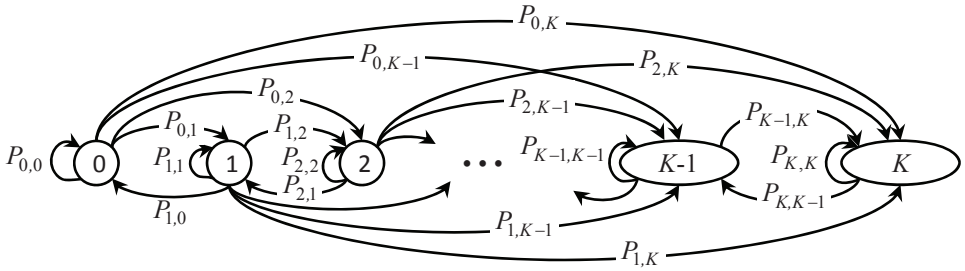


Figure 3.3: Markov model for the queueing behavior of a duty-cycling node without retransmission.

the probability that the observed node successfully sends a packet (all other nodes in the same grade pick a larger backoff counter) is

$$s_{n_i} = \sum_{w=0}^{W-1} \frac{1}{W} \left(\frac{W-w-1}{W} \right)^{n_i} ,$$

and the probability that the channel is sensed busy is

$$b_{n_i} = \sum_{w=0}^{W-1} \frac{1}{W} \left(1 - \left(\frac{W-w}{W} \right)^{n_i} \right) .$$

Then we obtain $p_t(i)$ and $p_s(i)$ as two functions of $p_e(i)$, respectively, i.e.,

$$p_t(i) = f(p_e(i)) = \sum_{n_i=0}^{N_i-1} C_{n_i} \cdot t_{n_i} , \tag{3.1}$$

$$p_s(i) = h(p_e(i)) = \sum_{n_i=0}^{N_i-1} C_{n_i} \cdot s_{n_i} , \tag{3.2}$$

$$p_c(i) = p_t(i) - p_s(i) , \tag{3.3}$$

$$p_b(i) = \sum_{n_i=0}^{N_i-1} C_{n_i} \cdot b_{n_i} . \tag{3.4}$$

3.3.3 Queueing Model without Retransmissions

Let random variable Q with value in $[0, K]$ denote the stationary queue length of an observed node when the node enters its `T` period of a cycle. Thus, as shown in Figure 3.3, the discrete-time Markov-chain model formed by Q has $K + 1$ states in total, each of which represents the number of packets in the queue at the start of the

node's T period. The queue length might be changed cycle by cycle, corresponding to the state transition in the Markov model. The transition interval with the cycle duration T_{cycle} is defined as the interval from the start of the current T period to the start of the next T.

During a cycle, for the observed node in grade i, 1) one packet is transmitted (i.e., removed from the queue) with the probability $p_t(i)$ if the queue is not empty; 2) k packets are generated with the probability $a_k = \frac{e^{-\lambda T_{cycle}}(\lambda T_{cycle})^k}{k!}$; and 3) one packet is successfully received with the probability $p_r(i)$. Then, the $(K+1) \times (K+1)$ state transition probability matrix $\mathbf{P}$ could be presented as

$$\mathbf{P} = \begin{bmatrix} P_{0,0} & P_{0,1} & \cdots & P_{0,K} \\ P_{1,0} & P_{1,1} & \cdots & P_{1,K} \\ \vdots & \vdots & \ddots & \vdots \\ P_{K,0} & P_{K,1} & \cdots & P_{K,K} \end{bmatrix} , \tag{3.5}$$

where $P_{m,n}$ is the transition probability that the queue length of the observed node changes from m to n,

$$\begin{cases} P_{0,0} & = a_0(1-p_r(i)), \\ P_{0,m} & = a_{m-1}p_r(i) + a_m(1-p_r(i)),\ m \in [1, K-1] \\ P_{m,n} & = 0,\ m \in [2, K],\ n \in [0, m-2] \\ P_{m,m-1} & = a_0 p_t(i)(1-p_r(i)),\ m \in [1, K] \\ P_{m,m} & = a_0[p_t(i)p_r(i) + (1-p_t(i))(1-p_r(i))] \\ & \quad + a_1 p_t(i)(1-p_r(i)),\ m \in [1, K-1] \\ P_{m,n} & = a_{n-m-1}(1-p_t(i))p_r(i) \\ & \quad + a_{n-m}[p_t(i)p_r(i) + (1-p_t(i))(1-p_r(i))] \\ & \quad + a_{n-m+1}p_t(i)(1-p_r(i)), \\ & \quad m \in [1, K-2],\ n \in [m+1, K-1] \\ P_{m,K} & = 1 - \sum_{n=0}^{K-1} P_{m,n},\ m \in [0, K] \end{cases} ,$$

and

$$p_r(i) = \eta_{i+1}(1-p_e(i+1))p_s(i+1) , \tag{3.6}$$

where $\eta_{i+1} \in [0, N_{i+1}]$ is the number of nodes in grade $i+1$ that set the observed node as the next-hop relaying node.

Let $\mathbf{B}(i) = [b_0(i),\ b_1(i),\ \cdots,\ b_K(i)]$ denote the stationary distribution of $b_q(i) =$

$\Pr\{Q = q\},\ (q \in [0, K])$. For the node in grade i, we have $\mathbf{B}(i) \times \mathbf{P} = \mathbf{B}(i)$ and $\sum_{q=0}^{K} b_q(i) = 1$. For the nodes in the highest grade, $p_r = 0$ since they cannot relay any packet for other nodes. Therefore, given that the packet generation information (λ and a_k) is known, there is only one undetermined variable in the transition matrix $\mathbf{P}$, i.e., p_t. Obviously, $p_e(i)$ is a function of $p_t(i)$, defined as

$$p_e(i) = b_0(i) = g(p_t(i)) \ . \tag{3.7}$$

By using numerical methods (e.g., the binary search algorithm), $p_t(i)$, $p_s(i)$, and $p_e(i)$ could be jointly determined by (3.1), (3.2), and (3.7). According to (3.6), $p_r(i-1)$ of the adjacent lower-grade nodes can be obtained with $p_s(i)$. Applying the same methods described above, we also obtain the corresponding $p_t(i-1)$, $p_s(i-1)$, and $p_e(i-1)$. Therefore, the related probabilities for the nodes in each grade can be calculated recursively.

3.3.4 Queueing Model with Retransmissions

One of the major differences of our model from [126] is that in this dissertation we consider the multi-hop transmission scenario where every node can generate packets. Specifically, let R denote the maximum number of retransmissions supported by the protocol, leading to $R+1$ retransmission stages (from zero to R) for a node. Therefore, the node state is determined by both of its retransmission stage and queue length. Figure 3.4 shows the 2-D Markov-chain queueing model with $(R+1)\cdot K+1$ states, each of which is represented by two indices, i.e., retransmission stage and queue length. For ease of drawing, note that in Figure 3.4, the states shown as dashed circles in the last row are the same as those in the first row.

As mentioned above, the probability that k packets are generated at a source node is a_k. Let $a_{\geq k}$ denote the probability that there are no less than k packets being generated in a cycle. Define $a_k = 0$ if $k < 0$, $a_{\geq k} = 1$ if $k \leq 0$, and $a_{\geq k} = 1 - \sum_{i=0}^{k-1} a_k$ if $k > 0$. We first examine the transitions from the empty-queue state $(0, 0)$, the probabilities of which only depend on the newly generated packets and the received packets from the neighbor sending node in the higher grade. Therefore, for a node in grade i, we have

$$P_{(0,0)\to(0,m)} = p_r(i)a_{m-1} + (1 - p_r(i))a_m,\ m \in [0, K-1] \tag{3.8}$$

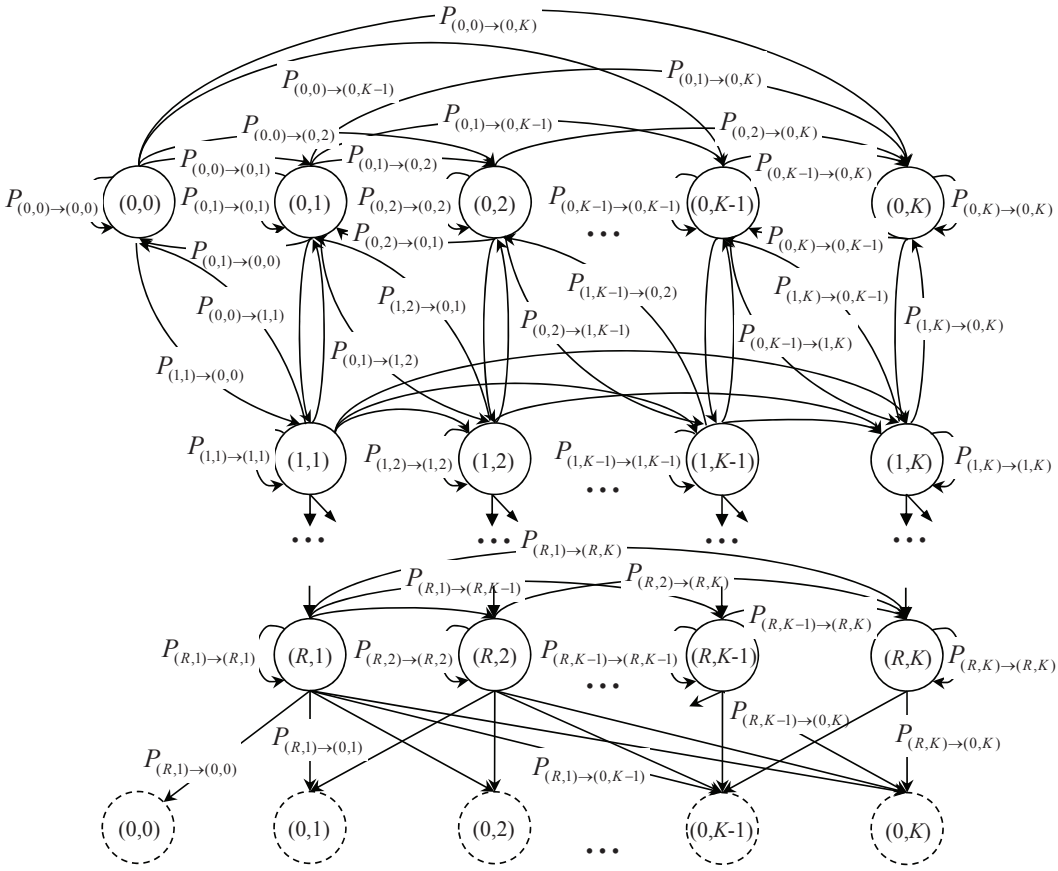


Figure 3.4: Markov model for the queueing behavior of a duty-cycling node with retransmission.

$$P_{(0,0)\to(0,K)} = p_r(i)a_{\geq K-1} + a_{\geq K} \ . \tag{3.9}$$

For the transitions between two states in the same retransmission stage, the corresponding transition probabilities are shown below. (3.10), (3.11) and (3.12) correspond to the transitions in retransmission stage 0, and (3.13) and (3.14) correspond to the transitions in a non-zero retransmission stage.

$$P_{(0,m)\to(0,m-1)} = p_s(1-p_r(i))a_0, \ m \in [1,K] \tag{3.10}$$

$$\begin{aligned} P_{(0,m)\to(0,n)} &= (1-p_t(i))p_r(i)a_{n-m-1} \\ &\quad + [p_s(i)p_r(i) + (1-p_t(i))(1-p_r(i))]a_{n-m} \\ &\quad + p_s(i)(1-p_r(i))a_{n-m+1}, \ m \in [1,K-1], \ n \in [m,K-1] \end{aligned} \tag{3.11}$$

$$\begin{aligned} P_{(0,m)\to(0,K)} &= (1-p_t(i))p_r(i)a_{\geq K-m-1} + [p_s(i)p_r(i)+ \\ &\quad (1-p_t(i))(1-p_r(i))]a_{\geq K-m} + (1-p_r(i))p_s(i)a_{\geq K-m+1}, \ m \in [1,K] \end{aligned} \tag{3.12}$$

$$P_{(r,m)\to(r,n)} = (1-p_t(i))p_r(i)a_{n-m-1} + (1-p_t(i))(1-p_r(i))a_{n-m}, \quad (3.13)$$
$$r \in [1,R],\ m \in [1,K-1],\ n \in [m,K-1]$$
$$P_{(r,m)\to(r,K)} = (1-p_t(i))p_r(i)a_{\geq K-m-1} + (1-p_t(i))(1-p_r(i))a_{K-m}, \quad (3.14)$$
$$r \in [1,R],\ m \in [1,K]$$

For the transitions between two states respectively in two adjacent retransmission stages, the transition probabilities are shown in (3.15) and (3.16). These transitions correspond to the cases where a retransmission is necessary after an RTS collision.

$$P_{(r,m)\to(r+1,n)} = p_c(i)p_r(i)a_{n-m-1} + p_c(i)(1-p_r(i))a_{n-m}, \quad (3.15)$$
$$r \in [0,R-1],\ m \in [1,K-1],\ n \in [m,K-1]$$
$$P_{(r,m)\to(r+1,K)} = p_c(i)p_r(i)a_{\geq K-m-1} + p_c(i)(1-p_r(i))a_{\geq K-m}, \quad (3.16)$$
$$r \in [0,R-1],\ m \in [1,K]$$

Finally, we examine the transitions from a non-zero retransmission stage to retransmission stage 0, corresponding to the cases either that a retransmitted packet is successfully delivered, as shown in (3.17) and (3.18), or that the packet is discarded when the retransmission limit is reached, as shown in (3.19) and (3.20).

$$P_{(r,m)\to(0,n)} = p_s(i)(1-p_r(i))a_{n-m+1} + p_s(i)p_r(i)a_{n-m}, \quad (3.17)$$
$$r \in [1,R-1],\ m \in [1,K],\ n \in [m-1,K-1]$$
$$P_{(r,m)\to(0,K)} = p_s(i)(1-p_r(i))a_{\geq K-m+1} + p_s(i)p_r(i)a_{\geq K-m}, \quad (3.18)$$
$$r \in [1,R-1],\ m \in [1,K]$$
$$P_{(R,m)\to(0,n)} = p_t(i)p_r(i)a_{n-m} + p_t(i)(1-p_r(i))a_{n-m+1}, \quad (3.19)$$
$$m \in [1,K],\ n \in [m-1,K-1]$$
$$P_{(R,m)\to(0,K)} = p_t(i)p_r(i)a_{\geq K-m} + p_t(i)(1-p_r(i))a_{\geq K-m+1},\ m \in [1,K] \quad (3.20)$$

For the transitions that are not listed above, the transition probabilities are zero.

Since the Markov chain is irreducible and aperiodic, for the node in grade i, the transition matrix $\mathbf{P}$ with states in the space $S = \{(0,0)\} \cup \{(m,n)|m = 0...R, n = 1...K\}$ has a unique stationary distribution $\mathbf{B}(i) = [b_{(0,0)}(i),\ b_{(0,1)}(i),\ \cdots,\ b_{(R,K)}(i)]$. Then we have $\mathbf{B}(i)\cdot\mathbf{P} = \mathbf{B}(i)$ and $\sum_{(m,n)\in S} b_{(m,n)}(i) = 1$. Similar to the model without retransmission, p_r for each grade is calculated recursively according to (3.6).

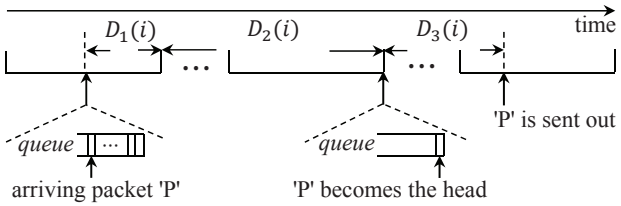
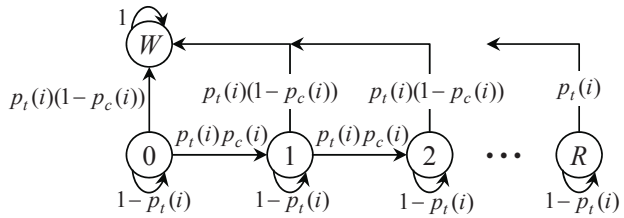
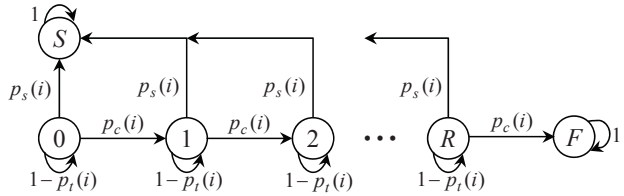


Figure 3.5: Three parts of the average delivery latency of a packet from its arriving at grade i to delivering to grade $i-1$ ($\sqcup$ stands for a cycle duration).



(a) For Calculating $D_2(i)$



(b) For Calculating $D_3(i)$

Figure 3.6: Absorbing Markov Chain for obtaining $D_2(i)$ and $D_3(i)$.

Therefore, with p_c known according to (3.3), and given that the packet generation information (λ, a_k and $a_{\geq k}$) is known, there are two undetermined variables in the transition matrix $\mathbf{P}$, i.e., p_t and p_s. Thus $p_e(i)$ is a function of $p_t(i)$ and $p_s(i)$,

$$p_e(i) = b_{(0,0)}(i) = G(p_t(i), p_s(i)) \ . \tag{3.21}$$

Combining (3.1), (3.2), and (3.21), $p_t(i)$, $p_s(i)$, and $p_e(i)$ can be obtained.

3.3.5 Performance Metrics

Average Packet Delivery Latency

With the probabilities obtained above, we derive the average latency of a packet sent from a source node in each grade to the sink node through multi-hop delivery with or without retransmissions. Specifically, the latency for a packet (the *observed packet*) from grade i ($i \geq 1$) being delivered to grade $i-1$ contains the following three

parts (see Figure 3.5):

(a) $D_1(i)$: the average waiting time from the time when the observed packet arrives in the current cycle to the beginning of the next cycle. If the node is the source that generated this packet, the average waiting time is $D_1(i) = D_{1s}(i) = \frac{T_{cycle}}{2}$. Otherwise, for a relaying node that receives this packet from grade $i+1$, the average waiting time is $D_1(i) = D_{1r}(i) = \frac{W}{2}\sigma + \text{SIFS} + durACK$.

(b) $D_2(i)$: the average waiting time from the beginning of the next cycle after the observed packet arrives in the current cycle to the beginning of a cycle when the packet becomes the head of the queue. It depends on the total queueing delay of the packets in front of the observed packet, including the packet service time of each head-of-queue packet, denoted by $d(i)$. For calculating $d(i)$, an absorbing Markov process [139] is shown in Figure 3.6a, where the number in the state shows the number of retransmissions that the packet has experienced, and *'W'* means that the packet wins the contention. Therefore,

$$d(i) = T_{cycle} \begin{cases} \mathcal{A}, & R = 0 \\ \mathcal{A}^{R+1} p_c(i)^R + \sum_{r=0}^{R-1} \mathcal{A}^{r+1} \left(1 - p_c(i)\right) p_c(i)^r, & R > 0 \end{cases} , \tag{3.22}$$

where $\mathcal{A} = \sum_{c_1=1}^{\infty} c_1 (1 - p_t(i))^{c_1 - 1} p_t(i) = \frac{1}{p_t(i)}$, c_1 is the number of cycles that the observed packet has experienced when it becomes the head-of-queue packet, and R is the maximum number of retransmissions the protocol can support. When $R = 0$, retransmission is not supported. According to the proposed Markov model,

$$D_2(i) = \sum_{k=0}^{K-1} \frac{d(i) \cdot k \cdot b_{Q,k}(i)}{1 - b_{Q,K}(i)} . \tag{3.23}$$

(c) $D_3(i)$: the service time of the observed packet from the beginning of a cycle when it just becomes the head of the queue to the time when it is sent successfully. Similarly, for $D_3(i)$, the absorbing Markov process of serving the observed packet is shown in Figure 3.6b, where *'S'* means that the packet is successfully sent out, and *'F'* means that the packet fails to be sent out and is removed from the queue. Therefore,

$$D_3(i) = T_{cycle} \sum_{r=0}^{R} \mathcal{B}^{r+1} p_c(i)^r p_s(i) , \tag{3.24}$$

where $\mathcal{B} = \sum_{c_2=0}^{\infty} c_2(1-p_t(i))^{c_2} = \frac{1-p_t(i)}{p_t(i)^2}$, and c_2 is the number of cycles that the observed packet has experienced when it is successfully sent out.

Then the average delivery latency of a packet sent from a source in grade h ($h \in [1, H]$) to the sink is

$$D_h = \sum_{i=1}^{h} [D_1(i) + D_2(i) + D_3(i)] \ . \tag{3.25}$$

Average Active Time Ratio per Cycle of a Node

The average active time per cycle of a node in grade i is composed of the average active time $E_i[R]$ in state `R` and the average active time $E_i[T]$ in state `T`. $E_i[R]$ depends on $p_r(i)$. Specifically, considering that a node needs to wait at least DIFS $+ W\sigma +$ *durRTS* time during its `R` state if it fails in receiving, we have $E_i[R] = p_r(i)(\frac{W}{2}\sigma + \sigma') + (1-p_r(i))(\text{DIFS} + W\sigma + durRTS)$, where $\sigma' = \text{DIFS} + durRTS + durCTS + durDATA + durACK + 3\text{SIFS}$. For $E_i[T]$, if the queue is empty, the active time during `T` state is zero; otherwise, there are three cases:

1. if the transmission is successful, the active time is $\Delta_s = \frac{W}{2}\sigma + \sigma'$,
2. if the transmission fails due to collision, the active time is $\Delta_c = \text{DIFS} + \frac{W}{2}\sigma + durRTS + \text{SIFS} + durCTS$,
3. if the channel is sensed busy, the active time is $\Delta_b = \text{DIFS} + \sum_{j=1}^{W-1} \frac{1}{W-1} \sum_{k=1}^{j} \frac{1}{j} \cdot (j-k+1) \cdot \sigma$.

Therefore, $E_i[T] = (1-p_e(i))[p_s(i)\cdot\Delta_s + p_c(i)\cdot\Delta_c + p_b(i)\cdot\Delta_b]$. Then the average active time ratio per cycle of a node in an H hop(s) chain network is $\Gamma_H = \frac{1}{H}\sum_{i=1}^{H} \frac{E_i[R]+E_i[T]}{T_{cycle}}$. This analysis is available for the models both with and without retransmission.

System Throughput

The system throughput S is defined as the number of packets received by the sink per second. In LSNs, the transmissions at the nodes 1-hop away from the sink determine the throughput. Thus $S = N_1 \cdot (1-p_e(1)) \cdot p_s(1)/T_{cycle}$, where N_1 is the number of nodes in grade one. The analysis is also available for both models.

Table 3.1: PDC Parameters

DIFS	10 ms	SIFS	5 ms
$durRTS$	11 ms	$durCTS$	11 ms
$durACK$	11 ms	durDATA	43 ms
W	16,32,64,128,256	σ	1 ms

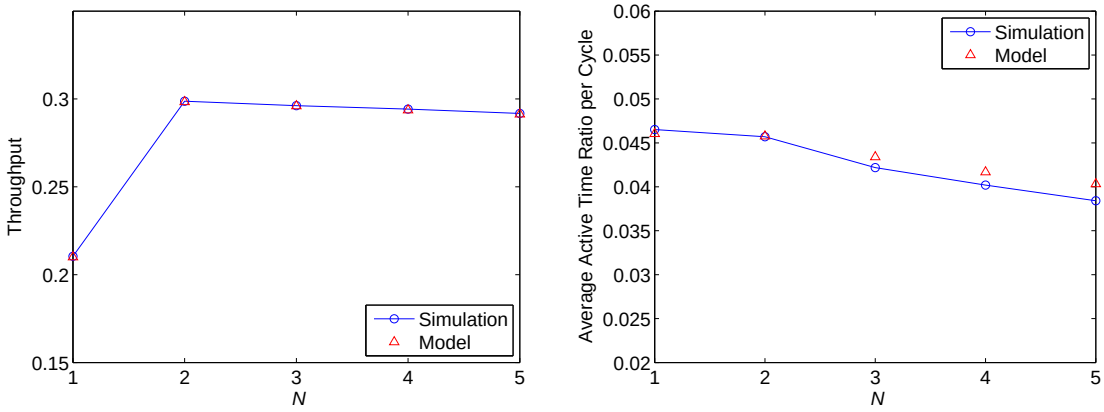
3.4 Validation for the Model without Retransmissions

In this section, we validate our proposed model without retransmission by comparing the analytical results derived in Section 3.3 with those obtained from the simulations based on OPNET[1]. The comparison is conducted under five scenarios by varying: 1) the number of nodes N in each grade, 2) the packet generation rate λ, 3) the sleep factor ξ, 4) the contention window size W, and 5) the queue capacity K. For simplicity, we assume that all grades contain the same number of node, and all nodes are set with the same λ, ξ, W and K. Note that it is not necessary for our model to follow these limits. The network is divided into $H = 7$ grades (therefore, there are $7 \times N$ sensor nodes in total in the network). Unless otherwise specified, we set $N = 3$, $\lambda = 0.03$, $\xi = 18$, $W = 64$, and $K = 15$. For each of the five scenarios, we vary one of these parameters and investigate the performance of PDC in terms of the system throughput, the average packet delivery latency from a node in nonzero grade to the sink node, and the active time ratio per cycle per node. The parameters used in PDC are shown in Table 3.1, which is the same as in [83]. Figures 3.7–3.11 show that our analytical results based on the proposed model match the simulation results quite well.

3.4.1 Varying the Number of Nodes in Each Grade

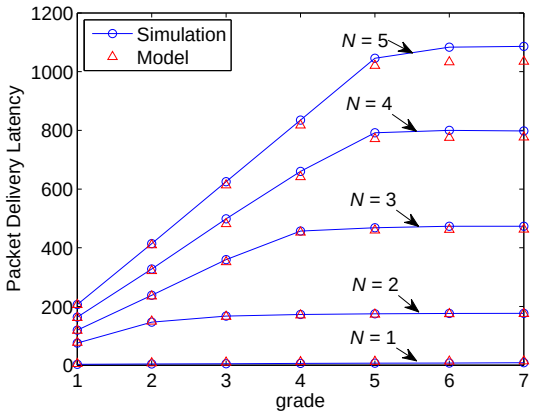
Figure 3.7 shows the performance results with N varying from one to five in each grade. In Figure 3.7a, the system throughput increases as N increases from one to two (correspondingly, the number of nodes in the network increases from seven to 14), and after then, it starts to decrease. Since all nodes in the network generate and send data to the sink node, the increase of nodes will contribute to the system throughput. However, as N becomes larger, there are more collisions occurring in the

[1]The code can be found in [140]



(a) Throughput (pkts/sec)

(b) Average Active Time Ratio



(c) Packet Delivery Latency (sec)

Figure 3.7: Performance with the varying number of nodes (N).

network, and as a result the system throughput slightly decreases.

The average active time ratio per cycle of a node is shown in Figure 3.7b, which indicates the energy consumption of the node. This average ratio is the total active time during a cycle of all nodes in the network divided by the total number of nodes. Because each time in the same grade, only one node can win the contention to transmit data and other failing nodes go to sleep to save energy, the ratio decreases as N increases. When $N = 2$, there is only one failing node each time, corresponding to a slight decrease of the active time ratio. As N becomes larger, there are more failing nodes in a grade, and thus the ratio decreases more.

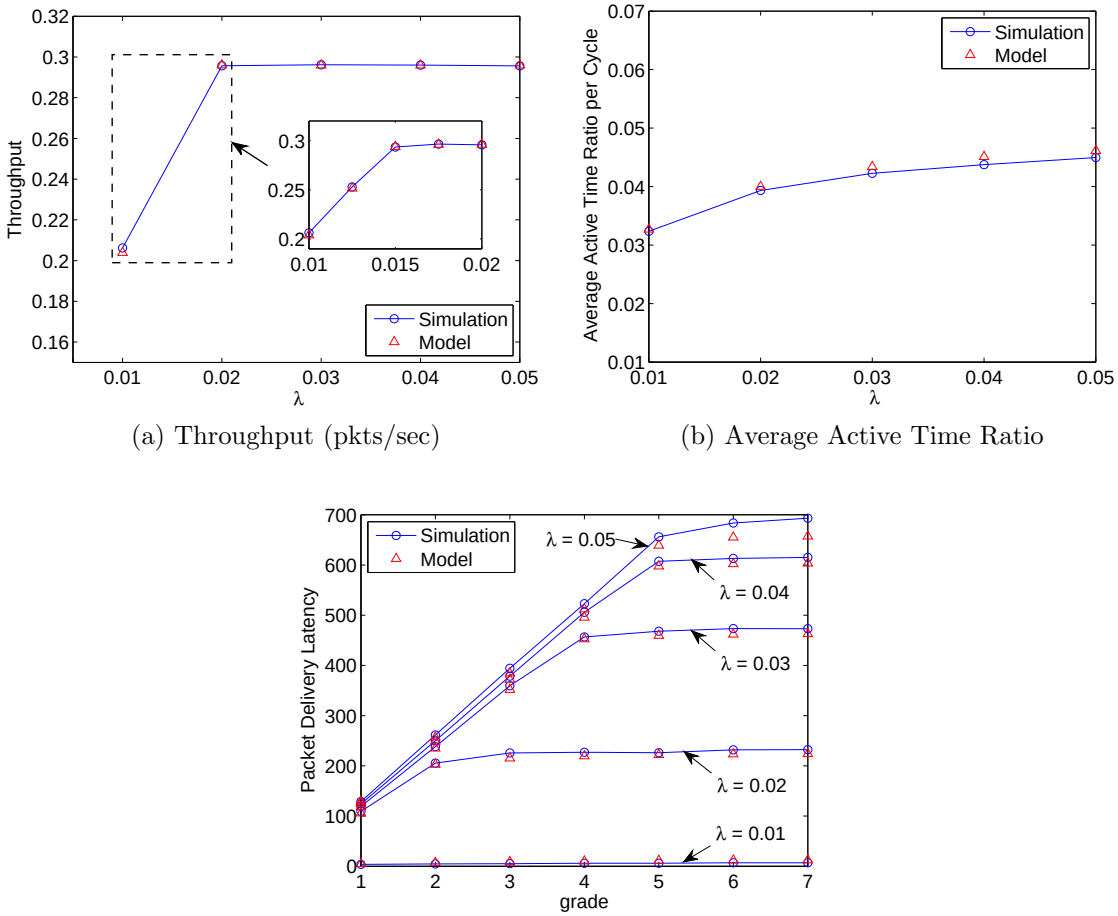
From Figure 3.7c, we observe that when there is only one node in each grade, a node can deliver the incoming packets as soon as they arrive at the node. Hence, there

are few packets accumulated in the queue and as a result, the packet delay is very short. Due to the pipelined-forwarding feature of PDC, the delay difference between the packets generated at the nodes farther away from the sink and those generated at the nodes closer to the sink is very small. However, when more nodes are added to the network as N increases, there are more packets accumulated in the queue due to a higher contention in the network. Therefore, the packet contending delay and queueing delay become longer, and as a result, the delivery latency to the sink is longer, too. We also observe that, as the grade increases, the packet delivery latency has a linear increase at first and then the increase becomes very slow. This is due to the fact that the traffic load on the nodes close to the sink is much heavier than that on the distant nodes. The long queueing delay in these nodes will significantly affect the delivery latency of the packets from the distant nodes. However, as the grade increases to a certain point, such an effect becomes weak and there is a turning point after which the delay increase is not so obvious. In addition, as N becomes larger, the turning point trends to appear at a higher grade.

3.4.2 Varying the Packet Generation Rate

In this scenario, we vary the packet generation rate λ at each node from 0.01 to 0.05 pkts/sec. Figure 3.8 shows the corresponding performance results. As shown in Figure 3.8a, when λ is smaller than the system service rate (which is $\frac{1}{H \cdot N \cdot T_{cycle}} \approx 0.014$), PDC can transmit packets as fast as they arrive at the network without overflowing the queues, and hence the system throughput increases linearly as the traffic load increases. However, the throughput remains the same as λ exceeds the system service rate and PDC saturates, i.e., each node always has a packet to send in every cycle and thus PDC reaches its delivery limit.

As λ increases, 1) the probability that a node has a packet to send in its active state increases, hence the active time ratio of each node increases as shown in Figure 3.8b and meanwhile the contention with other nodes in the same grade increases, leading to an increasing contending delay; 2) the average queue length increases and hence each packet has an increasing queueing delay. Therefore, as shown in Figure 3.8c, the average packet delivery latency at the nodes in each grade increases as λ increases. However, the amount of the increase decreases, since as λ increases, the average queue length approaches to the queue capacity, which limits the increase of the packet queueing delay and hence limits the packet delivery latency. In addition, for the same



(a) Throughput (pkts/sec)

(b) Average Active Time Ratio

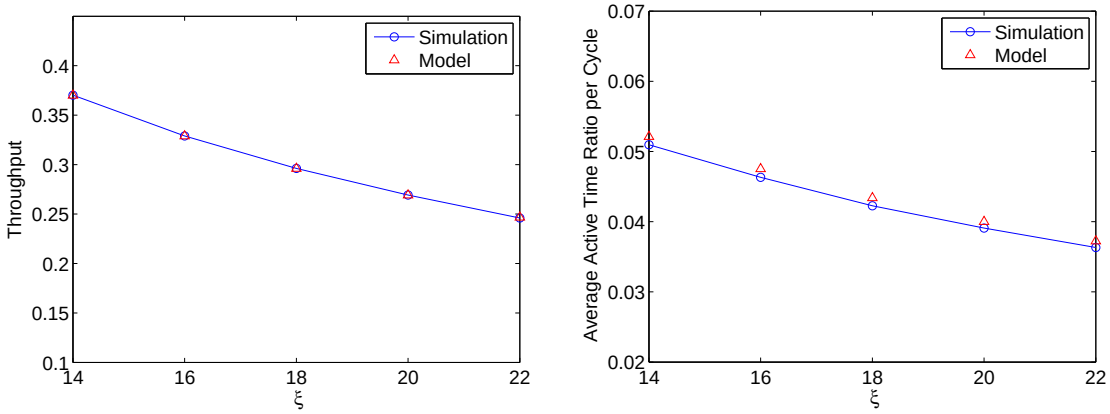
(c) Packet Delivery Latency (sec)

Figure 3.8: Performance with the varying packet generation rate (λ).

reason as shown in Figure 3.7c, there is a linear increase in the latency and then the increase is not so obvious as the grade increases, and the turning point trends to appear at a higher grade as λ becomes larger.

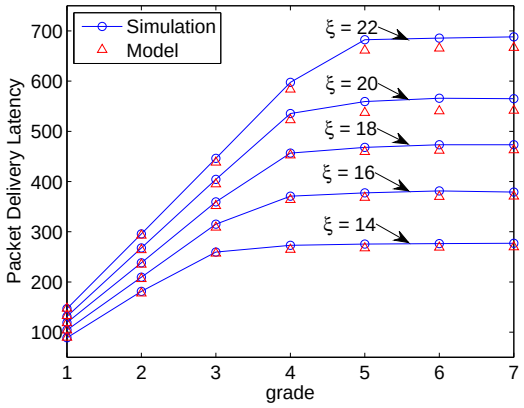
3.4.3 Varying the Sleep Factor

Figure 3.9 shows the performance results under the varying sleep factor ξ from 14 to 22. A larger sleep factor means a longer cycle duration. Since in PDC, at most one packet can be delivered per cycle, a longer cycle duration leads to a lower throughput. Therefore, the system throughput decreases as ξ increases, as shown in Figure 3.9a. For the average active time ratio, a longer cycle duration with a larger ξ leads to almost the same decrease trend as the throughput (see Figure 3.9b).



(a) Throughput (pkts/sec)

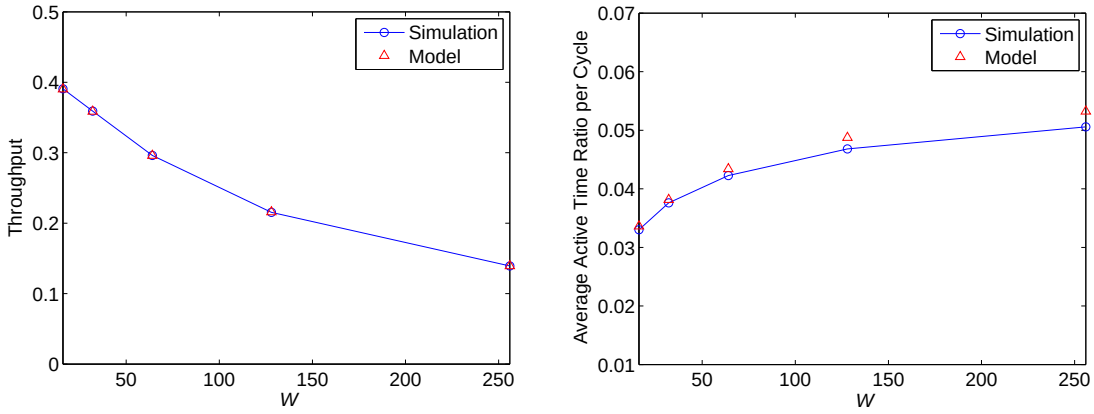
(b) Average Active Time Ratio



(c) Packet Delivery Latency (sec)

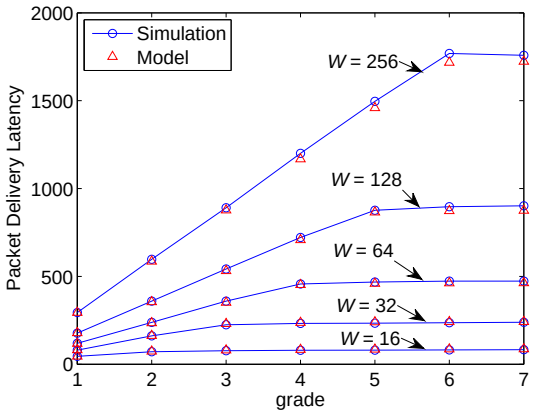
Figure 3.9: Performance with the varying sleep factor (ξ).

Note that the packet generation rate in this scenario is set to $\lambda = 0.03$, where PDC has already saturated for all of the sleep factors given according to both analytical and simulation results. All the nodes in the network can have a packet to send in every cycle. Therefore, the contention in the network remains almost constant as ξ increases. In other words, the increase of ξ does not affect the contending delay but does increase the queueing delay. From Figure 3.9c, we see that the average packet delivery latency at each grade increases almost linearly as ξ increases, since the queueing delay of packets increases linearly as the cycle duration increases.



(a) Throughput (pkts/sec)

(b) Average Active Time Ratio



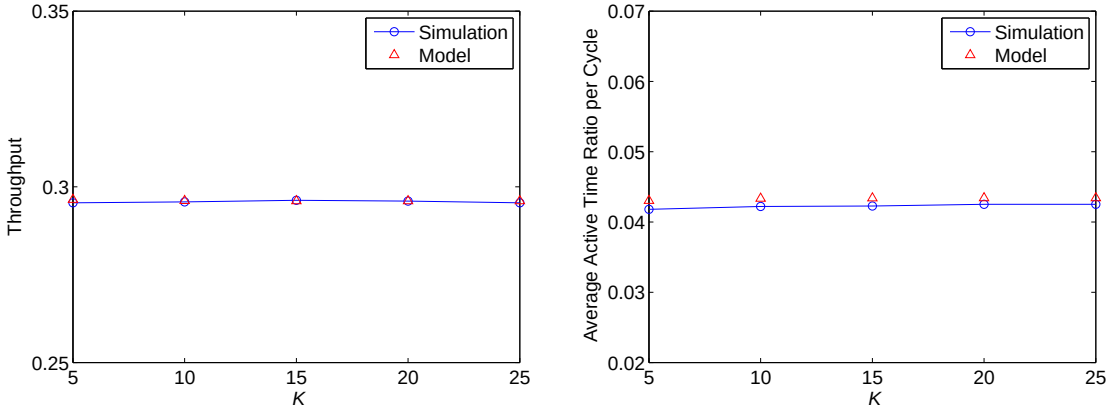
(c) Packet Delivery Latency (sec)

Figure 3.10: Performance with the varying contention window size (W).

3.4.4 Varying the Contention Window Size

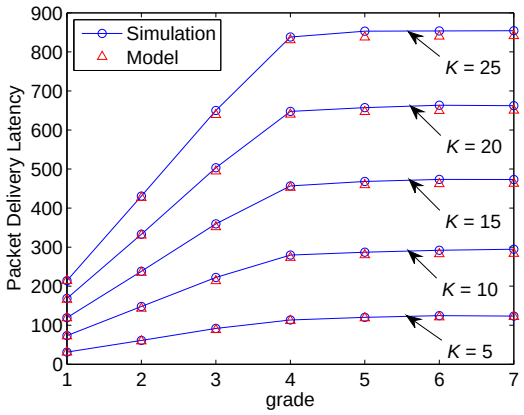
Figure 3.10 shows the performance results with the contention window size W exponentially varying from 16 to 256. Similar to ξ, a larger W means a longer cycle duration, which leads to a lower system throughput. Although increasing W decreases the probability of packet collision, which benefits the system throughput, such a decrease cannot dominate the variation of the throughput, as W is large enough when the total number of nodes that contend with each other at the same time is only $N = 3$. Since W increases exponentially, the system throughput also decreases exponentially, as shown in Figure 3.10a. On the contrary, the average active time ratio has an exponential increase as shown in Figure 3.10b.

Since the increase of the cycle duration has a dominant impact on the system in



(a) Throughput (pkts/sec)

(b) Average Active Time Ratio



(c) Packet Delivery Latency (sec)

Figure 3.11: Performance with the varying queue capacity (K).

comparison with the decrease of the packet collision probability as W increases, the queueing delay dominates the variation trend of the packet delivery latency. A larger W leads to a longer queueing delay and hence a longer packet delivery latency, as shown in Figure 3.10c. From Figure 3.10c, we also observe that the delivery latency at each grade has an exponential increase as W increases exponentially.

3.4.5 Varying the Queue Capacity

In this scenario, we vary the queue capacity K from five to 25 packets. Figure 3.11 shows the corresponding performance results. From Figure 3.11a, we see that the increase of K cannot benefit the system throughput, as PDC has already saturated with the settings of other four parameters (i.e., N, λ, ξ, and W) in this scenario.

Therefore, the system throughput keeps almost constant as K increases. So does the average active time ratio as shown in Figure 3.11b.

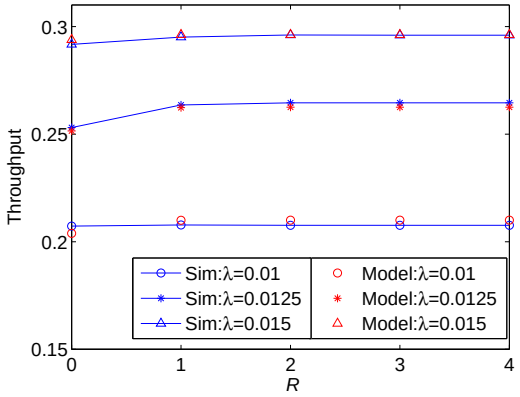
According to our analytical results, as $\lambda = 0.03$, the packet overflow happens with a certain probability at the nodes near the sink (e.g., about 0.3 at the grade-1 nodes). Therefore, the average queue length of these nodes increases as K increases, and hence the queueing delay increases. Since the contending delay keeps constant as PDC saturates, the packet delivery latency increases as the queueing delay increases. From Figure 3.11c, it shows that, as K increases linearly, the packet delivery latency at each grade increases linearly as well.

3.5 Validation for the Model with Retransmissions

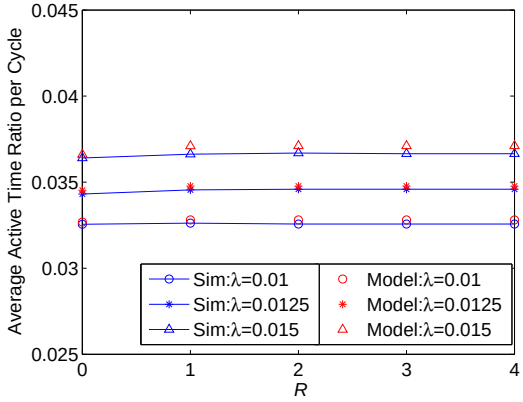
For validating the proposed model with the support of retransmission, we first set $N = 3$, $\xi = 18$, $W = 64$, and $K = 15$. λ is varied from 0.01 to 0.015 pkts/sec, and the maximum number of retransmissions R, that is supported by the protocol is varied from zero to four. Then we investigate the network performance in terms of throughput and average active time ratio, as shown in Figure 3.12a and Figure 3.12b, respectively, which indicates a good match with the simulation results.

From Figure 3.12a and Figure 3.12b, we observe that the variance of R almost has no impact on either throughput or active time ratio. This could be explained as follows. If a node is under a saturated condition such that it always has packets to send, retransmission will not improve throughput or change the active time ratio of a node. Conversely, if the node is unsaturated, it still has the capacity to retransmit a packet without causing overflow, which benefits the network performance. From another point of view, however, not being saturated also indicates that there are fewer packets experiencing collisions and requiring retransmissions. Therefore, the protocol with retransmissions almost has the same performance as that without retransmissions.

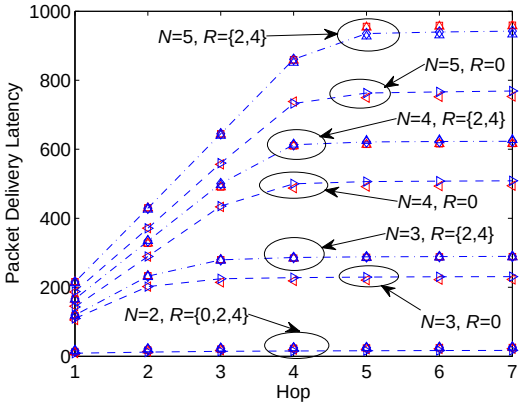
For evaluating the packet delivery latency, we set $W = 64$, $\lambda = 0.02$. N varies from two to five, and R varies from zero to four. The results are shown in Figure 3.12c, where the model results are shown in red curves, which match well with the simulation results shown in blue curves. The network is unsaturated when $N = 2$, since the system service rate $\frac{1}{H \cdot N \cdot T_{cycle}} \approx 0.022 > \lambda = 0.02$, in which collision can hardly happen and retransmission is not necessary. Thus whether retransmission is supported or not does not affect the packet delivery latency, as shown in the figure.



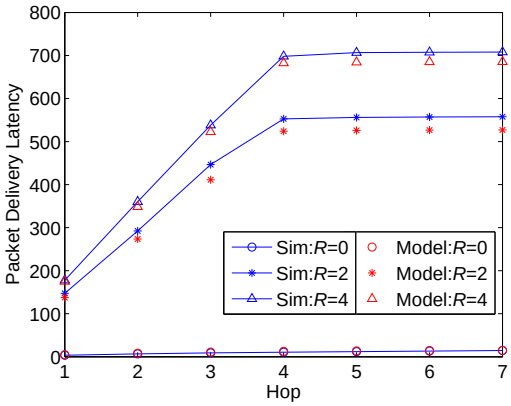
(a) Throughput ($N = 3$, $W = 64$)



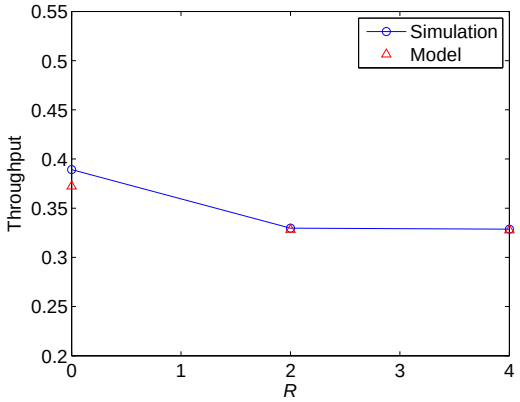
(b) Active Time Ratio ($N = 3$, $W = 64$)



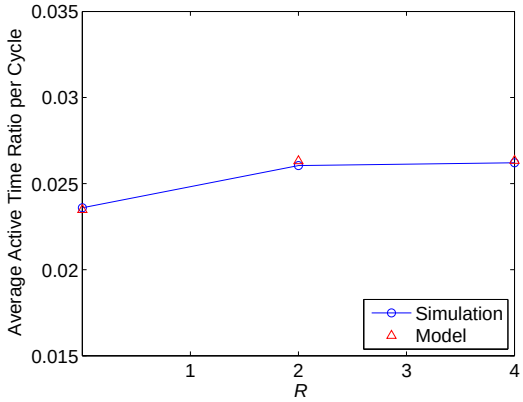
(c) Packet Delivery Latency ($N = \{2, 3, 4, 5\}$, $W = 64$, $\lambda = 0.02$)



(d) Packet Delivery Latency ($N = 5$, $W = 8$, $\lambda = 0.02$)



(e) Throughput ($N = 5$, $W = 8$, $\lambda = 0.02$)



(f) Active Time Ratio ($N = 5$, $W = 8$, $\lambda = 0.02$)

Figure 3.12: Performance with the support of retransmission.

On the other hand, for the saturated networks with $N = 3$ and 4, retransmission does increase the latency. However, it is interesting to see that in Figure 3.12c, the

latency almost has no change when R varies from two to four for all the networks with $N = 3, 4$ and 5. This is because that for all the three networks, the network contention level when $R = 2$ is the same as that when $R = 4$, and thus the average number of retransmissions that a packet experiences is no larger than two. If we change $W = 8$ to increase the network contention level for the network with $N = 5$, it shows that the variation of R from two to four increases the packet delivery latency, as shown in Figure 3.12d. The network throughput and node active time ratio are also shown in Figure 3.12e and Figure 3.12f, respectively. As shown in the figures, the network throughput decreases and the node active time ratio increases when retransmission is supported, but the variation of R from two to four still has almost no impact on them, which has been analyzed above.

Considering that the data redundancy issue is common to see in sensor networks [141], and that the variation of R has very slight impact on either the network throughput or energy efficiency, whereas a larger R may lead to a larger packet delivery latency, we show that retransmission will not benefit much in a linear sensor network.

3.6 Conclusions

In this chapter, we applied a duty-cycled protocol with the pipelined-scheduling (i.e., PDC) to an LSN, modeled the MAC process with and without retransmission, and analyzed its performance in the LSN in terms of the system throughput, network energy efficiency and packet delivery latency. The model has been validated through the extensive OPNET simulations. By modifying its first part, i.e., the MAC process analysis, the model can be easily generalized to other duty-cycled MACs. We have utilized the model to analyze the network performance under various system parameters.

Chapter 4

Approach to Ran2Ran NDD

4.1 Overview

As mentioned before, Nodal Distance Distribution (NDD) can be utilized to characterize most of the performance metrics in WANETs due to their nonlinear relationships with the distances among nodes. In this chapter, we propose a systematic and algorithmic approach to Ran2Ran NDDs. Based on the approach, we first obtain the NDDs associated with arbitrary triangles, which are further utilized to obtain the NDDs with arbitrary polygons, since any polygons can be triangulated.

4.2 Related Work

To achieve analytical tractability with the stochastic geometry tools, most researchers assumed that an infinite number of nodes at any time instant are spatially distributed as a homogeneous PPP over an infinite area [14–16, 68, 69, 142], which gives rise to a stationary location-independent performance as seen from the perspective of each node in the network. Such a PPP model is extremely useful because it can obtain some insightful results with the tractable mathematical expressions that can vastly simplify the performance analysis by using Campbell's theorem [13].

However, the PPP model is inadequate/inaccurate in many practical wireless networks where a finite number of nodes are distributed at random in a finite area, because it assumes an unbounded number of nodes and does not take into account the effects of the network boundaries. For example, in order to utilize the PPP model to obtain the distribution of the distance to the n-nearest neighbor for ad hoc and

sensor networks, both the network area and the number of sensor nodes were assumed to be infinite in [143], while in reality they are finite. As an another example, with the stochastic geometry tools in cellular networks, BSs were usually assumed to be deployed randomly following a PPP model in an infinite area with all cells working on the same frequency band [68,69], whereas in reality BSs are very possibly deployed according to cell planning and their locations are not totally random. Especially, in a cellular system with a higher frequency reuse factor, e.g., seven, where there is no interference from adjacent cells and thus it only needs to investigate the performance metrics of a single cell, or with a nearby, nonadjacent cell, or in a sector-partitioned cell, where the BS is equipped with multiple directional antennas working on different frequency bands [67] and thus it is of interest to study the performance metrics of a single sector, the results based on PPP are not applicable directly.

As a result, the approaches based on a BPP had been developed in [17–19,144,145] for the modeling and analysis of a finite wireless network, called binomial network where a finite number of nodes are independently and uniformly distributed and communicating/interfering with a reference receiving node. With the developed BPP model, Srinivasa et al. [144] analyzed the interference seen at the reference receiver located at the center of a d-dimensional ball network. The work was then extended in [17] where the reference receiver could be arbitrarily located within the network. Torrieri et al. [18] analyzed the BPP outage probability of the reference receiver which is located at the origin of a circular network, while Guo et al. [145] and Valenti et al. [19] investigated the BPP outage at an arbitrarily-located reference receiver within an arbitrarily-shaped network. Although the BPP model can handle finite wireless networks, it cannot analyze the exterior interference at a reference receiver, i.e., the reference receiver and the interfering transmitters are not in the same network. For example, in an uplink scenario of a cellular system with a frequency reuse factor of one, it is worth investigating the interference at the BS of a cell from the transmitting cellular users in its neighbor cells [146]; in a two-tier cellular system, the analysis of the mutual interference at the BSs between femto and macro cells is of significant importance [20]; and in a finite wireless network with all nodes uniformly distributed at random, it is also significant to analyze the average performance metrics at any node instead of at a specific reference node [22, 27, 48], where the BPP model is inapplicable.

The limitation that the exterior interference at a reference receiver in a finite wireless network cannot be analyzed with the BPP models can be relaxed by using the

probabilistic distance-based models with appropriate Ref2Ran NDDs. For instance, in [47], the NDDs from a BS (as the reference node) to an arbitrary cellular user (as the random node) in the same hexagonal cell or to the users in its adjacent cells were derived, based on which the received exterior co-channel interference at a BS from a transmitting cellular user of any six adjacent cells was explicitly analyzed. The obtained NDDs were further utilized in [146] to analyze the cumulative exterior interference at a BS from its all first-layer neighbor cells (note that all cells in the above cellular systems use the same frequency band).

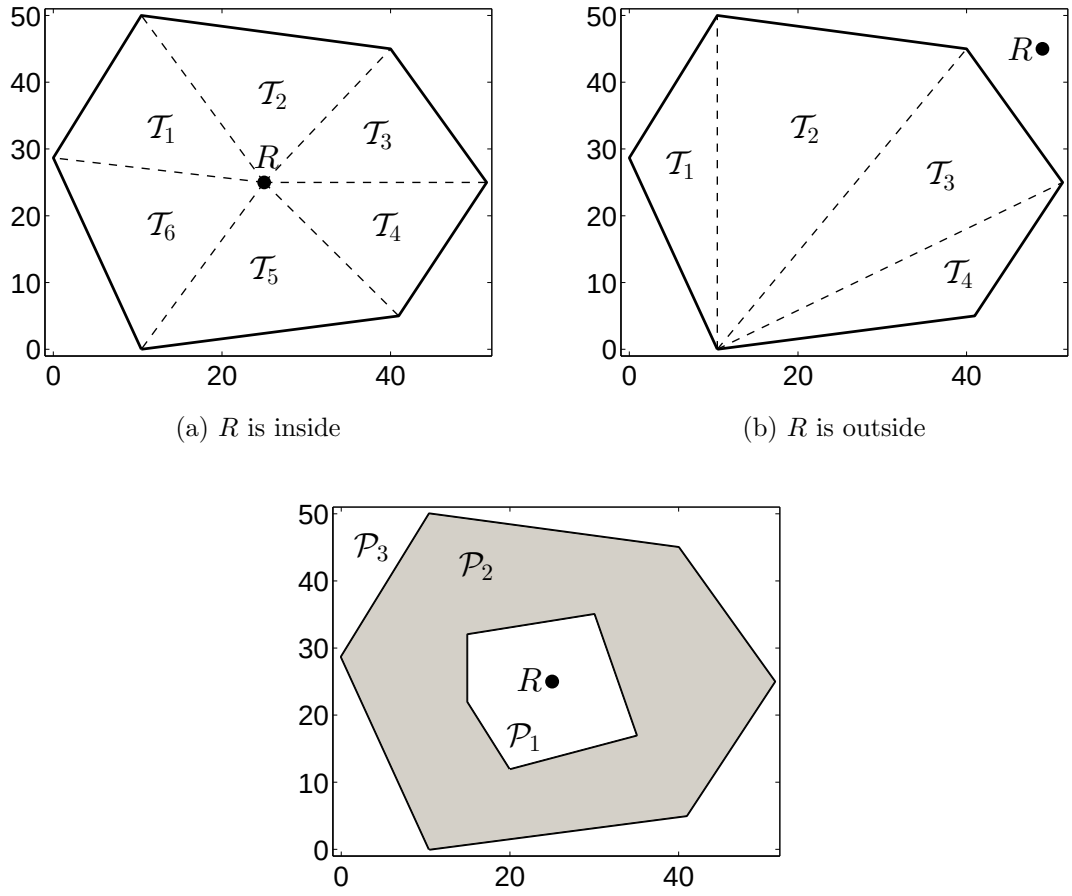
Recently, there has been a breakthrough made in obtaining the Ref2Ran NDDs from an arbitrary reference node (i.e., anywhere in the plane) to a random node inside an arbitrary polygon [58], which eliminates the inappropriate assumptions and limitations that the network area has to be in certain specific shapes (including squares [42], disks/circles [23], hexagons [43,47], regular polygons [54], and convex n-gons [53]) and the reference node has to be inside or on the boundary of the network. Specifically, we first obtain the Ref2Ran NDD from a vertex of an arbitrary triangle to the triangle using the area-ratio approach, and then based on which the Ref2Ran NDD from an exterior or interior reference point to the triangle can be obtained through Decomposition and Recursion (D&R) methods. Such NDDs from an arbitrary reference point to an arbitrary triangle are called Ref2Ran triangle-NDDs.

Based on the obtained Ref2Ran triangle-NDDs, the distribution of the distance from an arbitrary reference point to an arbitrary polygon can also be obtained through a D&R method, since any polygon can be triangulated. If a reference point R is inside a polygon, the polygon can be triangulated from R as shown in Figure 4.1a. On the other hand, if R is outside, we can just triangulate the polygon as shown in Figure 4.1b. Then with a weighted probabilistic sum, the CDF of the distance distribution from R to the polygon is

$$F(d) = \sum_{i}^{K} \frac{S_i}{S} F_i(d) \ , \tag{4.1}$$

where K is the number of triangles generated after the triangulation ($K = 6$ and 4 for the examples shown in Figure 4.1a and Figure 4.1b, respectively), S_i is the area of triangle $\mathcal{T}_i$, S is the area of the polygon, and $F_i(d)$ is the CDF of the distance distribution from R to $\mathcal{T}_i$, which has been obtained already.

It is possible that a subarea of a network area has a higher node density than



(a) R is inside

(b) R is outside

(c) Tiered structure

Figure 4.1: An arbitrary reference point R and arbitrary polygons (unit: m).

other subareas, which is referred to as tiered network structure, due to the reasons such as the bottleneck nodes near the hotspot under heavy loads run out of energy, some nodes are physically damaged in a hostile environment, or there are no nodes being deployed in a specific area which needs not to be monitored. In this case, the above weighted probabilistic sum is still applicable, but with the weights modified correspondingly due to the node density difference. Take the tiered structure shown in Figure 4.1c for example. Assuming the node density ratio between $\mathcal{P}_1$ (the white area) and $\mathcal{P}_2$ (the gray area) is $\lambda_1 : \lambda_2$ (λ_1 and λ_2 are not zero at the same time), then the distance distribution from an arbitrary reference point R to the whole area $\mathcal{P}_3$ ($\mathcal{P}_1$ plus $\mathcal{P}_2$) is

$$F_3(d) = \sum_{i}^{2} \frac{S_i \lambda_i}{\sum_j^2 S_j \lambda_j} F_i(d) \ , \tag{4.2}$$

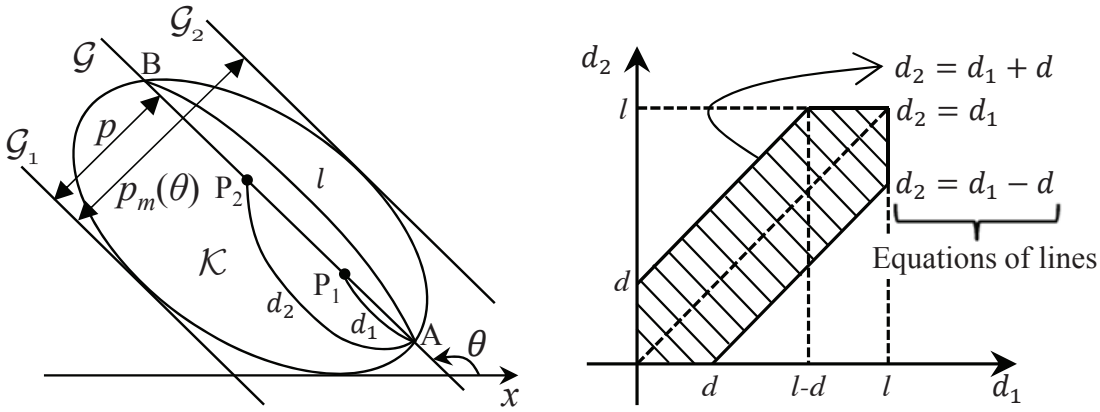
where S_i is the area of $\mathcal{P}_i$, and $F_i(d)$ is the distance distribution from R to $\mathcal{P}_i$, which can be obtained by (4.1). The obtained results have been applied in our recent work [20] to analyze the outage probability of the macro and femto BSs in arbitrarily-shaped cells for tiered cellular networks.

The authors in [59] made an algorithmic implementation based on our proposed approach. Instead of using D&R methods, they modified the shoelace formula to calculate the area of the intersection between the polygon and the circle centered at an arbitrary reference point R with a radius of d. Then the probability that the distance from R to a point uniformly distributed at random within the polygon is no longer than d is the area of the intersection divided by the area of the polygon. In the modified shoelace formula, the area of the intersection between the circle and each triangle generated by triangulating the polygon from R is obtained based on our approach.

On the other hand, for analyzing the finite wireless networks where both of the communicating nodes are randomly distributed, the Ran2Ran NDDs are required to build the probabilistic distance-based models. The Ran2Ran NDDs associated with disks [46, 51, 60], triangles [52, 55], rectangles [22, 24, 25, 51], rhombuses [44], trapezoids [50], and regular polygons [22, 27, 45, 48, 49, 56] had been obtained. The corresponding approaches can be categorized into standard statistics [27], quadratic product [48], and area ratio [22]. The recent breakthrough is to obtain the explicit Ran2Ran NDDs within an arbitrary convex geometry based on Chord Length Distribution (CLD). Specifically, let $\mathcal{K}$ denote a convex geometry in the plane and $\mathcal{G}$ a random line intersecting with $\mathcal{K}$. The chord length $C = |\mathcal{K} \cap \mathcal{G}|$ is a random variable, known as the *random chord* of $\mathcal{K}$. Let D denote the distance between two points independently and uniformly distributed within $\mathcal{K}$. The relationship between CLD of $\mathcal{K}$ and the Ran2Ran NDD within $\mathcal{K}$ is

$$f_D(d) = \frac{1}{S^2} \int_d^{c_m} 2d(c-d)u f_C(c)\, \mathrm{d}c \,, \tag{4.3}$$

where $f_C(c)$ and $f_D(d)$ are the PDFs of C and D, respectively, c_m is the maximum chord length, and S and u are the area and perimeter of $\mathcal{K}$, respectively. Bäsel derived the CLDs for regular polygons [56] and right-angled triangles [52], based on which the corresponding Ran2Ran NDDs were also obtained according to (4.3). Recently, we derived the Ran2Ran NDDs associated with arbitrary triangles [55] based on the corresponding CLDs obtained in [147]. Ren [148] obtained the CLDs



(a) A convex geometry (b) The integral domain of $F_{\mathcal{G}}(d)$ in (4.5)

Figure 4.2: Ran2Ran NDD within a convex network, $\mathcal{K}$.

for some special convex polygons (e.g., equilateral triangles, rectangles, and regular hexagons). However, it becomes too complicated to obtain the CLD for an arbitrary convex n-gon ($n \geq 4$) as n becomes larger. Moreover, the approach cannot obtain the Ran2Ran NDDs associated with concave, disjoint, or tiered network regions such as clusters/cells in finite wireless networks, since the chord definition cannot be applied directly.

4.3 Approach to Ran2Ran NDD

As aforementioned, the constraints of the existing approaches to Ran2Ran NDDs lead to their limited applicability in modeling and analyzing finite wireless networks. In this section, we propose a systematic and algorithmic approach based on Kinematic Measure (KM) in integral geometry [61, 149] to obtain the Ran2Ran NDDs for arbitrarily-shaped wireless networks, including convex, concave, disjoint, and tiered networks.

4.3.1 Convex Network Regions

As shown in Figure 4.2a, let $\mathcal{K}$ denote an arbitrary convex network with area of S in the two-dimensional plane, and $\mathcal{G}$ a line uniformly distributed at random in the plane and intersecting with $\mathcal{K}$, which produces a chord with length of $|AB| = l$. Without Loss Of Generality (WLOG), define the orientation of $\mathcal{G}$ as θ ($0 \leq \theta \leq \pi$) which is the angle made by the part of the line $\mathcal{G}$ above x-axis with the positive direction of the x-

axis. There are two points, P_1 and P_2, uniformly distributed at random on the chord AB. $|AP_1| = d_1$ and $|AP_2| = d_2$ $(0 \leq d_1, d_2 \leq l)$. Define set $\mathcal{H} = \{\mathcal{G}|\mathcal{G} \cap \mathcal{K} \neq \emptyset\}$ and $\mathcal{B}_{\mathcal{G}}(d) = \{(d_1, d_2)|\ |P_1P_2| = |d_1 - d_2| \leq d, P_1, P_2 \in \mathcal{G} \cap \mathcal{K}\}$. According to the KM concept in integral geometry, it is easy to know that the probability that the distance (denoted as a random variable D) between any two points which are uniformly distributed at random within $\mathcal{K}$ is no larger than d is

$$F_D(d) = \Pr(|P_1P_2| \leq d) = \frac{1}{S^2} \int_{\mathcal{H}} \int_{\mathcal{B}_{\mathcal{G}}(d)} \mathrm{d}_{P_1} \mathrm{d}_{P_2} = \frac{1}{S^2} \int_{\mathcal{H}} \int_{\mathcal{B}_{\mathcal{G}}(d)} |d_1 - d_2|\ \mathrm{d}d_1 \mathrm{d}d_2\ \mathrm{d}\mathcal{G}\ . \tag{4.4}$$

Denote the inner integral $\int_{\mathcal{B}_{\mathcal{G}}(d)} |d_1 - d_2|\ \mathrm{d}d_1 \mathrm{d}d_2$ as $F_{\mathcal{G}}(d)$. With the integral domain of $F_{\mathcal{G}}(d)$ shown in Figure 4.2b, it is easy to know that

$$F_{\mathcal{G}}(d) = \int_{\mathcal{B}_{\mathcal{G}}(d)} |d_1 - d_2|\ \mathrm{d}d_1 \mathrm{d}d_2 = \iint\limits_{|d_1 - d_2| \leq d} |d_1 - d_2|\ \mathrm{d}d_1 \mathrm{d}d_2 = d^2 l - \frac{2}{3} d^3\ , \tag{4.5}$$

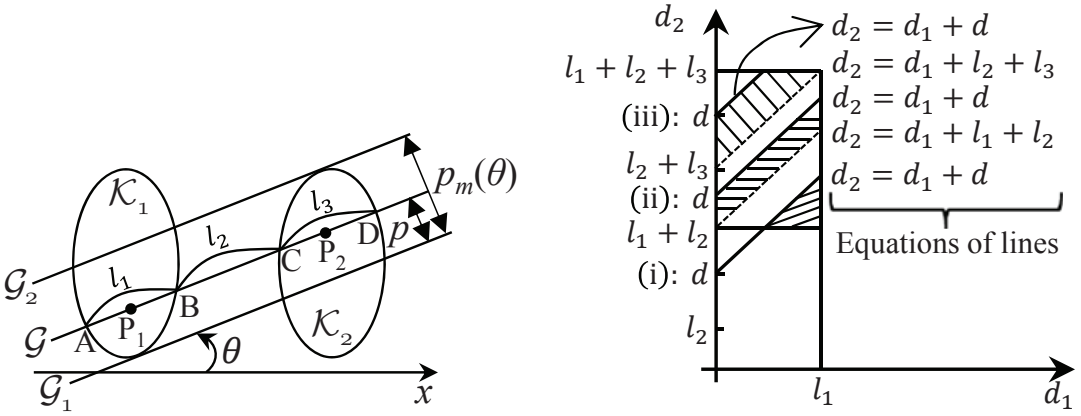
where l is determined by $\mathcal{G}$. For the outer integral, there are two tangents with the same orientation θ, i.e., $\mathcal{G}_1$ and $\mathcal{G}_2$ as shown in Figure 4.2a, which completely encompass $\mathcal{K}$ and are known as support lines in integral geometry. Therefore, a line $\mathcal{G}$ intersecting with $\mathcal{K}$ is determined by θ and its distance p to, WLOG, $\mathcal{G}_1$ $(0 \leq p \leq p_m(\theta)$, where $p_m(\theta)$, a function of θ, is the distance between the two support lines). Thus, l in $F_{\mathcal{G}}(d)$ is a function of θ and p (in what follows, we still use l for the ease of presentation). Finally, (4.4) can be written as

$$F_D(d) = \frac{1}{S^2} \int_0^{\pi} \int_0^{p_m(\theta)} F_{\mathcal{G}}(d)\ \mathrm{d}p \mathrm{d}\theta\ . \tag{4.6}$$

Let $f_{\mathcal{G}}(d) = \frac{\mathrm{d}F_{\mathcal{G}}(d)}{\mathrm{d}d} = 2d(l - d)$, then the PDF of D is

$$f_D(d) = \frac{1}{S^2} \int_0^{\pi} \int_0^{p_m(\theta)} f_{\mathcal{G}}(d)\ \mathrm{d}p \mathrm{d}\theta\ . \tag{4.7}$$

The closed-form expression of $f_D(d)$ depends on a specific network shape. Section 4.4 will show in detail how to obtain the closed-form Ran2Ran NDDs associated with arbitrary triangles, and based on which how to obtain the Ran2Ran NDDs associated with arbitrary polygons.



(a) Disjoint network regions

(b) The integral domains for each $F_{\mathcal{G}}(d)$ in three cases: (i) $l_2 \le d \le l_1 + l_2$; (ii) $l_1 + l_2 \le d \le l_2 + l_3$; (iii) $l_2 + l_3 \le d \le l_1 + l_2 + l_3$

Figure 4.3: Ran2Ran NDD between two disjoint network regions, $\mathcal{K}_1$ and $\mathcal{K}_2$.

4.3.2 Concave or Disjoint Network Regions

The above approach only applies to convex networks. However, for the Ran2Ran NDDs associated with concave or disjoint networks, it has to consider the random distance between two disjoint regions. Take the case shown in Figure 4.3a as an example, where there are two disjoint geometries in the two-dimensional plane, i.e., $\mathcal{K}_1$ and $\mathcal{K}_2$ with area of S_1 and S_2, respectively. $\mathcal{G}$ is a line with orientation θ uniformly distributed at random in the plane and intersecting with both $\mathcal{K}_1$ and $\mathcal{K}_2$, which produces two chords with length of $|AB| = l_1$ and $|CD| = l_3$, inside $\mathcal{K}_1$ and $\mathcal{K}_2$, respectively, and a segment with length of $|BC| = l_2$ outside. $\mathcal{G}_1$ and $\mathcal{G}_2$ are the two lines (also called support lines in this dissertation) which are parallel with $\mathcal{G}$, intersect with both geometries, and tangent to $\mathcal{K}_1$ and $\mathcal{K}_2$, respectively. Therefore, $\mathcal{G}$ is also determined by θ and p ($0 \le p \le p_m(\theta)$), where p is the distance from $\mathcal{G}$ to one of the support lines (e.g., $\mathcal{G}_1$ in Figure 4.3a) and $p_m(\theta)$ is the distance between the two support lines. Similarly, l_1, l_2, and l_3 are also determined by (θ, p)[1]. P_1 and P_2 are the two random points uniformly distributed on AB and CD, respectively. $|AP_1| = d_1$, $|AP_2| = d_2$, and $|P_1P_2| = d_2 - d_1$. Obviously, $0 \le d_1 \le l_1$ and $l_1 + l_2 \le d_2 \le l_1 + l_2 + l_3$. To obtain $F_D(d) = \Pr(|P_1P_2| \le d)$, the key is to obtain $F_{\mathcal{G}}(d)$ according to (4.6). WLOG, assume $l_1 \le l_3$ and thus $l_1 + l_2 \le l_2 + l_3$ (the analysis is the same for $l_3 \le l_1$). There are three cases with respect to the range of d for obtaining $F_{\mathcal{G}}(d)$.

[1]Likewise in the following, unless explicitly stated otherwise, l_i ($i \in \{1, 2, 3\}$) is equivalent to $l_i(\theta, p)$ for the ease of presentation.

The corresponding integral domains for each case are shown in Figure 4.3b. Therefore,

(i) $l_2 \leq d \leq l_1 + l_2$:

$$F_{\mathcal{G}}^{\mathrm{i}}(d) = \iint\limits_{d_2 - d_1 \leq d} (d_2 - d_1)\ \mathrm{d}d_1\mathrm{d}d_2 = \frac{(l_2 - d)^2(l_2 + 2d)}{6}\ ,$$
$$f_{\mathcal{G}}^{\mathrm{i}}(d) = \frac{\mathrm{d}F_{\mathcal{G}}^{\mathrm{i}}(d)}{\mathrm{d}d} = (d - l_2)d\ .$$

(ii) $l_1 + l_2 \leq d \leq l_2 + l_3$:

$$F_{\mathcal{G}}^{\mathrm{ii}}(d) = \frac{l_1}{2}(d + l_1 + l_2)(d - l_1 - l_2)\ ,$$
$$f_{\mathcal{G}}^{\mathrm{ii}}(d) = \frac{\mathrm{d}F_{\mathcal{G}}^{\mathrm{ii}}(d)}{\mathrm{d}d} = l_1 d\ .$$

(iii) $l_2 + l_3 \leq d \leq l_1 + l_2 + l_3$:

$$F_{\mathcal{G}}^{\mathrm{iii}}(d) = \frac{1}{6}(2l_2 + 2l_3 + d)(l_2 + l_3 - d)^2 - \frac{1}{2}\left((l_2 + l_3)^2 - d^2\right)(l_1 + l_2 + l_3 - d)\ ,$$
$$f_{\mathcal{G}}^{\mathrm{iii}}(d) = \frac{\mathrm{d}F_{\mathcal{G}}^{\mathrm{iii}}(d)}{\mathrm{d}d} = (l_1 + l_2 + l_3 - d)d\ .$$

On the other hand, for the case that P_1 is on chord CD and P_2 on chord AB, the same results can be obtained. With a probabilistic sum, we have

$$F_{\mathcal{G}}(d) = \begin{cases} 2F_{\mathcal{G}}^{\mathrm{i}}(d), & l_2 \leq d \leq l_1 + l_2 \\ 2F_{\mathcal{G}}^{\mathrm{ii}}(d), & l_1 + l_2 \leq d \leq l_2 + l_3 \\ 2F_{\mathcal{G}}^{\mathrm{iii}}(d), & l_2 + l_3 \leq d \leq l_1 + l_2 + l_3 \end{cases}\ . \tag{4.8}$$

By substituting (4.8) and $S_1 S_2$ for the $F_{\mathcal{G}}(d)$ and S^2 in (4.6), respectively, the CDF of the random distance between two disjoint arbitrarily-shaped geometries can be obtained. Note that for the two separate geometries sharing a common boundary, l_2 is zero if the line $\mathcal{G}$ intersects with the both two geometries through the boundary, so (4.8) can still be utilized.

The approach can be further extended to obtain the distribution of the distance between two points uniformly distributed at random within a concave geometry. As shown in Figure 4.4, there are two cases associated with the segment in the line passing through the two random points and between them: case 1) the segment is

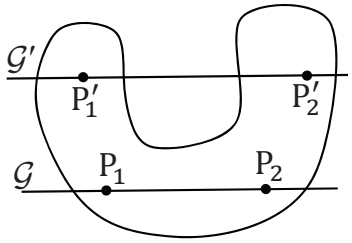
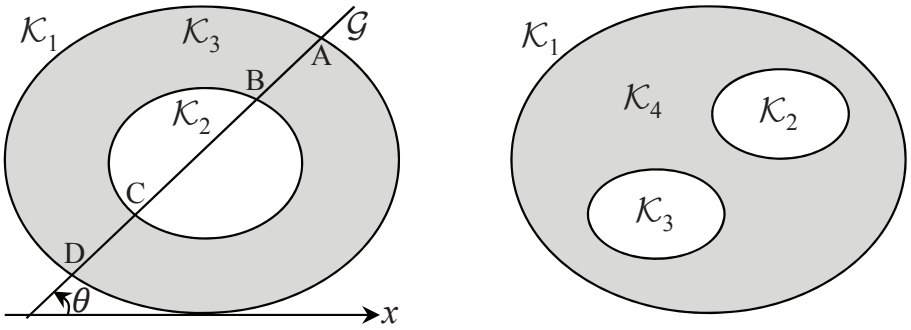


Figure 4.4: Ran2Ran NDD associated with concave geometry.



(a) The network $\mathcal{K}_1$ contains another network $\mathcal{K}_2$

(b) The network $\mathcal{K}_1$ contains two networks $\mathcal{K}_2$ and $\mathcal{K}_3$

Figure 4.5: Ran2Ran NDD associated with tiered networks.

completely contained in the geometry, e.g., the segment along line $\mathcal{G}$ in Figure 4.4; case 2) the segment cannot be completely contained, e.g., the segment along line $\mathcal{G}'$ in the figure. By applying (4.5) to obtain $F_{\mathcal{G}}(d)$ for case 1) and (4.8) for case 2), the intended concave distance distribution can be obtained according to (4.6).

4.3.3 Tiered Network Regions

The proposed approach is further extended to obtain the Ran2Ran NDDs for the tiered networks. As an example shown in Figure 4.5a, a network area $\mathcal{K}_1$ contains another network $\mathcal{K}_2$ inside (could be a void subregion of the network, i.e., a hole area with no nodes deployed). Denote the gray region inside $\mathcal{K}_1$ but outside $\mathcal{K}_2$ as $\mathcal{K}_3$. With (4.6), we obtain the CDF of the random distances within $\mathcal{K}_1$ and that within $\mathcal{K}_2$, denoted as $F_{11}(d)$ and $F_{22}(d)$, respectively (hereafter, we use $F_{xx}(d)$ to denote the CDF of the random distances within $\mathcal{K}_x$, and $F_{xy}(d)$ or $F_{yx}(d)$ the CDF of the random distances between $\mathcal{K}_x$ and $\mathcal{K}_y$). Nevertheless, the Ran2Ran NDD in $\mathcal{K}_3$ (i.e., $F_{33}(d)$) and that between two different regions (i.e., $F_{12}(d)$, $F_{13}(d)$, and $F_{23}(d)$), are also of great interest and very useful for analyzing some network scenarios where existing

approaches cannot apply. To this end, we first show how to obtain $F_{23}(d)$ by using the approach to obtaining the CDF of the distance between two disjoint geometries as illustrated in Figure 4.3a (a special case where l_2 is always zero). The others can then be obtained by a weighted probabilistic sum with decomposition and recursion.

As shown in Figure 4.5a, a line $\mathcal{G}$ with orientation θ and intersecting with $\mathcal{K}_2$ produces three segments, i.e., AB and CD inside $\mathcal{K}_3$, and BC inside $\mathcal{K}_2$. According to (4.8), one can find, say $F^a_{\mathcal{G}}(d)$, associated with the distance between AB and BC, and $F^b_{\mathcal{G}}(d)$ associated with the distance between CD and BC. Then,

$$F_{23}(d) = \frac{1}{S_2 S_3} \int_0^{\pi} \int_0^{p_m(\theta)} \left(F^a_{\mathcal{G}}(d) + F^b_{\mathcal{G}}(d) \right) \, \mathrm{d}p\mathrm{d}\theta \ , \tag{4.9}$$

where S_x is the area of $\mathcal{K}_x$. Meanwhile, with a weighted probabilistic sum,

$$F_{11}(d) = \frac{S_2}{S_1} \underbrace{\left(\frac{S_2}{S_1} F_{22}(d) + \frac{S_3}{S_1} F_{23}(d) \right)}_{F_{12}(d)} + \frac{S_3}{S_1} \underbrace{\left(\frac{S_2}{S_1} F_{23}(d) + \frac{S_3}{S_1} F_{33}(d) \right)}_{F_{13}(d)} \ . \tag{4.10}$$

With $F_{11}(d)$ and $F_{22}(d)$, and $F_{23}(d)$ obtained based on (4.6) and (4.9), respectively, $F_{12}(d)$, $F_{13}(d)$, and $F_{33}(d)$ can be obtained through (4.10). There are some remarks regarding (4.10). First, for obtaining $F_{11}(d)$ with the weighted probabilistic sum, each of the two random nodes could be in either $\mathcal{K}_2$ or $\mathcal{K}_3$. If one of them is in $\mathcal{K}_2$, the other one could be in either $\mathcal{K}_2$ or $\mathcal{K}_3$, and thus (4.10) contains the first part, i.e., $\frac{S_2}{S_1}(\frac{S_2}{S_1}F_{22}(d) + \frac{S_3}{S_1}F_{23}(d))$. Similarly, if one of them is in $\mathcal{K}_3$, the other one could also be in either $\mathcal{K}_2$ or $\mathcal{K}_3$, so the second part, $\frac{S_3}{S_1}(\frac{S_2}{S_1}F_{23}(d) + \frac{S_3}{S_1}F_{33}(d))$, is also in (4.10). Second, the obtained $F_{33}(d)$ is actually the Ran2Ran NDD within the gray area $\mathcal{K}_3$ shown in Figure 4.5a (the area of $\mathcal{K}_1$ minus $\mathcal{K}_2$). Instead of using the probabilistic sum method as shown above, $F_{33}(d)$ can also be obtained by applying the approach introduced in Section 4.3.2 for concave or disjoint geometries. With $F_{11}(d)$, $F_{22}(d)$, and $F_{33}(d)$ obtained, $F_{23}(d)$ can be obtained by using (4.10) instead of (4.9).

Note that if there are nodes uniformly deployed in $\mathcal{K}_2$ and $\mathcal{K}_3$ but with different node densities, the above weighted probabilistic sum is still applicable, but with the weights modified correspondingly due to the node density differences. Specifically, assuming the node density ratio between $\mathcal{K}_2$ and $\mathcal{K}_3$ is $\lambda_2 : \lambda_3$ (λ_2 and λ_3 are not zero

at the same time),

$$\begin{aligned} F_{11}(d) &= \sum_{i=2}^{3}\sum_{j=2}^{3}\frac{S_i d_i S_j d_j}{(\sum_{k=2}^{3} S_k\lambda_k)^2}F_{ij}(d)\ , \\ F_{1i}(d) &= \sum_{j=2}^{3}\frac{S_j\lambda_j}{\sum_{k=2}^{3} S_k\lambda_k}F_{ij}(d)\ , (i\in\{2,3\})\ . \end{aligned} \tag{4.11}$$

Especially, if $\lambda_2 : \lambda_3 = 1 : 1$, i.e., all areas have the same node density, then $F_{11}(d)$ and $F_{1i}(d)$ $(i \in \{2,3\})$ are what shown in (4.10). Therefore, the proposed approach in this dissertation has a potential of handling the networks with nonuniform nodal distributions.

The approach can be easily generalized to multi-network cases. For example, there are two networks in $\mathcal{K}_1$, labeled by $\mathcal{K}_2$ and $\mathcal{K}_3$, respectively, as shown in Figure 4.5b. The gray area (labeled by $\mathcal{K}_4$) plus $\mathcal{K}_2$ and that plus $\mathcal{K}_3$ are labeled by $\mathcal{K}_5$ and $\mathcal{K}_6$, respectively (not shown in the figure). Based on the above approaches, $F_{22}(d)$, $F_{33}(d)$, $F_{23}(d)$, $F_{26}(d)$, and $F_{35}(d)$ can be obtained directly. Assuming the node density among $\mathcal{K}_2$, $\mathcal{K}_3$, and $\mathcal{K}_4$ is $\lambda_2 : \lambda_3 : \lambda_4$ (λ_2, λ_3, and d_4 are not zero at the same time),

$$\begin{aligned} F_{26}(d) &= \sum_{i=3}^{4}\frac{S_i\lambda_i}{\sum_{k=3}^{4} S_k\lambda_k}F_{2i}(d)\ , \\ F_{35}(d) &= \sum_{i=2,4}\frac{S_i\lambda_i}{\sum_{k=2,4} S_k\lambda_k}F_{3i}(d)\ , \\ F_{11}(d) &= \sum_{i=2}^{4}\sum_{j=2}^{4}\frac{S_i\lambda_i S_j\lambda_j}{(\sum_{k=2}^{4} S_k\lambda_k)^2}F_{ij}(d)\ , \end{aligned} \tag{4.12}$$

based on which $F_{24}(d)$, $F_{34}(d)$, and $F_{44}(d)$ can be obtained.

4.4 Ran2Ran NDD Associated with Arbitrary Polygons

Since most of the existing work on the modeling and analysis of wireless communication networks approximated the network shapes as polygons [20, 22–25, 27, 29, 42, 43, 47, 146, 150, 151], this section will focus on obtaining the Ran2Ran NDDs associated with arbitrary polygons. As shown in Figure 4.6, we first apply our approach to obtain the Ran2Ran NDDs associated with arbitrary triangles (i.e., Ran2Ran triangle-NDDs), including the Ran2Ran NDD within a triangle, and that between any two triangles which can be disjoint or share either a common vertex or side. The

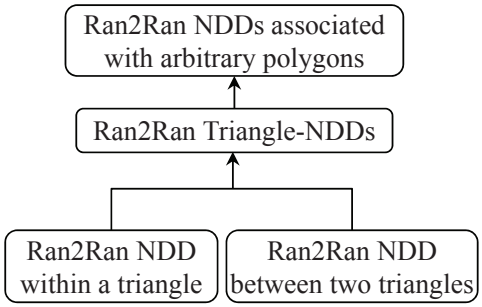


Figure 4.6: Obtain the Ran2Ran NDDs for the arbitrarily-shaped polygonal networks based on the Ran2Ran triangle-NDDs.

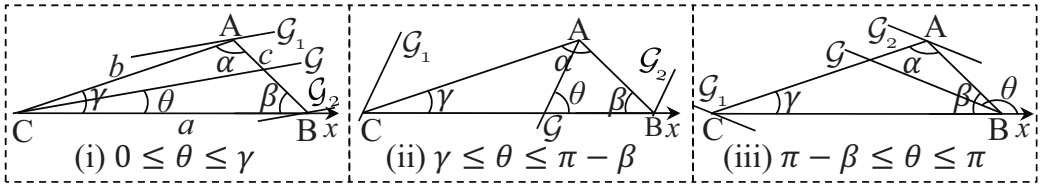


Figure 4.7: Ran2Ran NDD within an arbitrary triangle.

obtained Ran2Ran triangle-NDDs can be utilized for modeling and analyzing the wireless communication networks associated with triangle geometries, such as sensor networks with triangle-shaped clusters [152] and triangle-shaped cellular systems with highly directional antennas [153]. Furthermore, based on the obtained Ran2Ran triangle-NDDs, we show how to obtain the Ran2Ran NDDs associated with arbitrary polygons (including the polygons with several smaller polygons embedded inside, forming a tiered structure) through the decomposition and recursion methods, since any polygons can be triangulated. Matlab simulations are conducted to verify all of the obtained results.

4.4.1 Ran2Ran NDD within a Triangle

$\triangle ABC$ is an arbitrary triangle with side lengths $|CB| = a$, $|AC| = b$, and $|AB| = c$, internal angles $\angle A = \alpha$, $\angle B = \beta$, and $\angle C = \gamma$, and area $||\triangle ABC|| = S$. WLOG, $a \ge b \ge c$, and let side CB be on x-axis, as shown in Figure 4.7. For simplicity, assume $a = 1$ and other edges are normalized correspondingly. The PDF of the Ran2Ran NDD, $f_D(d)$, can be scaled to any size of triangles with $a = s$ by

$$f_{sD} = \frac{1}{s} f_D(\frac{d}{s}) \ , \tag{4.13}$$

where f_{sD} is the corresponding PDF of the Ran2Ran NDD of the triangle with $a = s$. Such a scaling is applicable to any polygons. According to (4.7), when calculating the length of the chord produced by a line intersecting with the triangle with orientation θ with regard to x-axis, there are three cases in terms of the range of θ: (i) $0 \leq \theta \leq \gamma$, (ii) $\gamma \leq \theta \leq \pi - \beta$, (iii) $\pi - \beta \leq \theta \leq \pi$. A systematic and algorithmic procedure is designed for the numerical integration of (4.7), based on which the corresponding closed-form expression is also derived, as shown below in detail.

Specifically, let θ increase from zero to π with a fixed small step of $\delta\theta$ (e.g., $\delta\theta = \frac{\pi}{180}$). For each θ with a set of lines intersecting with the triangle, $\mathcal{G}$ is the line which produces the longest chord of length $b(\theta)$. The distance between the two tangents parallel with $\mathcal{G}$, i.e., $\mathcal{G}_1$ and $\mathcal{G}_2$ which completely encompass the whole triangle, is $p_m(\theta)$. The distance between $\mathcal{G}_1$ and $\mathcal{G}$ and that between $\mathcal{G}_2$ and $\mathcal{G}$ are $p_1(\theta)$ and $p_2(\theta)$, respectively. Obviously, $p_m(\theta) = p_1(\theta) + p_2(\theta)$. With p increasing from 0 to $p_m(\theta)$ with a fixed small step δp (e.g., $\delta p = \frac{1}{1{,}000}$), $\frac{p_m(\theta)}{\delta p}$ chords are generated. For each chord of length l calculated based on trigonometry, $f_{\mathcal{G}}(d)$ is obtained according to (4.5), based on which the PDF of the Ran2Ran NDD can be obtained according to (4.7).

Taking the case (i) (shown in Figure 4.7) where $0 \leq \theta \leq \gamma$ for example, the following is to show how to obtain the corresponding closed-form expression. Specifically, for $0 \leq p \leq p_1(\theta)$, a chord is determined by (θ, p) with length of $l = \frac{p \cdot b(\theta)}{p_1(\theta)}$. From $l \geq d$, the integration range of p is $[\frac{d \cdot p_1(\theta)}{b(\theta)}, p_1(\theta)]$, and $\theta \leq \theta_1^{\mathrm{i}} = \arcsin\left(\frac{b\sin(\alpha)}{d}\right) - \beta$ or $\theta \geq \theta_2^{\mathrm{i}} = \pi - \arcsin\left(\frac{b\sin(\alpha)}{d}\right) - \beta$. Meanwhile, for $p_1(\theta) \leq p \leq p_m(\theta)$, the length of the determined chord is $l = \frac{(p_m(\theta) - p) \cdot b(\theta)}{p_2(\theta)}$; and from $l \geq d$, the integration range of $p \in [p_1(\theta), p_m(\theta) - \frac{d \cdot p_2(\theta)}{b(\theta)}]$, and $\theta \leq \theta_1^{\mathrm{i}}$ or $\theta \geq \theta_2^{\mathrm{i}}$. Therefore,

$$f_D^{\mathrm{i}}(d) = \begin{cases} f_1^{\mathrm{i}} & \gamma \leq \frac{\pi}{2} - \beta \\ f_{21}^{\mathrm{i}} + f_{22}^{\mathrm{i}} & \text{otherwise} \end{cases}, \tag{4.14}$$

where

$$f_1^{\mathrm{i}} = \begin{cases} H_1^{\mathrm{i}}\left(0, \theta_1^{\mathrm{i}}\right) + H_2^{\mathrm{i}}\left(0, \theta_1^{\mathrm{i}}\right) & 0 \leq \theta_1^{\mathrm{i}} \leq \gamma \\ H_1^{\mathrm{i}}(0, \gamma) + H_2^{\mathrm{i}}(0, \gamma) & \theta_1^{\mathrm{i}} > \gamma \\ 0 & \text{otherwise} \end{cases},$$

$$f_{21}^{\mathrm{i}} = \begin{cases} H_1^{\mathrm{i}}\left(0, \theta_1^{\mathrm{i}}\right) + H_2^{\mathrm{i}}\left(0, \theta_1^{\mathrm{i}}\right) & 0 \leq \theta_1^{\mathrm{i}} \leq \frac{\pi}{2} - \beta \\ 0 & \text{otherwise} \end{cases},$$

$$f_{22}^{\mathrm{i}} = \begin{cases} H_1^{\mathrm{i}}\left(\theta_2^{\mathrm{i}}, \gamma\right) + H_2^{\mathrm{i}}\left(\theta_2^{\mathrm{i}}, \gamma\right) & \theta_2^{\mathrm{i}} \leq \gamma \\ 0 & \text{otherwise} \end{cases},$$

$$H_1^{\mathrm{i}}(\mathcal{X}, \mathcal{Y}) = \frac{1}{S^2} \int_{\mathcal{X}}^{\mathcal{Y}} \int_{\frac{d \cdot p_1(\theta)}{b(\theta)}}^{p_1(\theta)} 2d\left(l_1 - d\right) \, \mathrm{d}p\mathrm{d}\theta \,, \tag{4.15}$$

$$H_2^{\mathrm{i}}(\mathcal{X}, \mathcal{Y}) = \frac{1}{S^2} \int_{\mathcal{X}}^{\mathcal{Y}} \int_{p_1(\theta)}^{p_m(\theta) - \frac{d \cdot p_2(\theta)}{b(\theta)}} 2d\left(l_2 - d\right) \, \mathrm{d}p\mathrm{d}\theta \,. \tag{4.16}$$

Similarly, for case (ii),

$$f_D^{\mathrm{ii}}(d) = f_1^{\mathrm{ii}} + f_2^{\mathrm{ii}} \,, \tag{4.17}$$

where

$$f_1^{\mathrm{ii}} = \begin{cases} H_1^{\mathrm{ii}}(\gamma, \theta_1^{\mathrm{ii}}) + H_2^{\mathrm{ii}}(\gamma, \theta_1^{\mathrm{ii}}) & \gamma \leq \theta_1^{\mathrm{ii}} \leq \frac{\pi}{2} \\ 0 & \text{otherwise} \end{cases},$$

$$f_2^{\mathrm{ii}} = \begin{cases} H_1^{\mathrm{ii}}(\theta_2^{\mathrm{ii}}, \pi - \beta) + H_2^{\mathrm{ii}}(\theta_2^{\mathrm{ii}}, \pi - \beta) & \theta_2^{\mathrm{ii}} \leq \pi - \beta \\ 0 & \text{otherwise} \end{cases},$$

$$\theta_1^{\mathrm{ii}} = \arcsin\left(\frac{c\sin(\beta)}{d}\right) \,,$$

$$\theta_2^{\mathrm{ii}} = \pi - \arcsin\left(\frac{c\sin(\beta)}{d}\right) \,,$$

and for case (iii),

$$f_D^{\mathrm{iii}}(d) = \begin{cases} f_1^{\mathrm{iii}} & \beta \leq \frac{\pi}{2} - \gamma \\ f_{21}^{\mathrm{iii}} + f_{22}^{\mathrm{iii}} & \text{otherwise} \end{cases}, \tag{4.18}$$

where

$$f_1^{\mathrm{iii}} = \begin{cases} H_1^{\mathrm{iii}}(\pi - \beta, \pi) + H_2^{\mathrm{iii}}(\pi - \beta, \pi) & \theta_1^{\mathrm{iii}} < \pi - \beta \\ H_1^{\mathrm{iii}}(\theta_1^{\mathrm{iii}}, \pi) + H_2^{\mathrm{iii}}(\theta_1^{\mathrm{iii}}, \pi) & \pi - \beta \leq \theta_1^{\mathrm{iii}} \leq \pi \\ 0 & \text{otherwise} \end{cases},$$

$$f_{21}^{\mathrm{iii}} = \begin{cases} H_1^{\mathrm{iii}}(\pi - \beta, \theta_2^{\mathrm{iii}}) + H_2^{\mathrm{iii}}(\pi - \beta, \theta_2^{\mathrm{iii}}) & \pi - \beta \leq \theta_2^{\mathrm{iii}} \leq \frac{\pi}{2} + \gamma \\ 0 & \text{otherwise} \end{cases},$$

$$f_{22}^{\mathrm{iii}} = \begin{cases} H_1^{\mathrm{iii}}(\theta_1^{\mathrm{iii}}, \pi) + H_2^{\mathrm{iii}}(\theta_1^{\mathrm{iii}}, \pi) & \theta_1^{\mathrm{iii}} \leq \pi \\ 0 & \text{otherwise} \end{cases},$$

$$\theta_1^{\mathrm{iii}} = \pi - \arcsin(\tfrac{c\sin(\alpha)}{d}) + \gamma,$$

$$\theta_2^{\mathrm{iii}} = \arcsin(\tfrac{c\sin(\alpha)}{d}) + \gamma.$$

Finally, the PDF of Ran2Ran NDD within an arbitrary triangle is

$$f_D(d) = f_D^{\mathrm{i}}(d) + f_D^{\mathrm{ii}}(d) + f_D^{\mathrm{iii}}(d) \,. \tag{4.19}$$

Similar to H_1^{i} and H_2^{i}, H_1^{ii} and H_1^{iii} can be calculated using (4.15), and H_2^{ii} and H_2^{iii} can be calculated using (4.16), with different $p_1(\theta)$, $p_2(\theta)$ and $b(\theta)$ for different cases. The closed-form expressions of $H_1(\mathcal{X},\mathcal{Y})$ and $H_2(\mathcal{X},\mathcal{Y})$ for each case are shown below,

$$\begin{cases} H_1^{\mathrm{i}}(\mathcal{X},\mathcal{Y}) = h_1^{\mathrm{i}}(\mathcal{Y}) - h_1^{\mathrm{i}}(\mathcal{X}) \ , H_2^{\mathrm{i}}(\mathcal{X},\mathcal{Y}) = h_2^{\mathrm{i}}(\mathcal{Y}) - h_2^{\mathrm{i}}(\mathcal{X}) \ , \\ H_1^{\mathrm{ii}}(\mathcal{X},\mathcal{Y}) = h_1^{\mathrm{ii}}(\mathcal{Y}) - h_1^{\mathrm{ii}}(\mathcal{X}) \ , H_2^{\mathrm{ii}}(\mathcal{X},\mathcal{Y}) = h_2^{\mathrm{ii}}(\mathcal{Y}) - h_2^{\mathrm{ii}}(\mathcal{X}) \ , \\ H_1^{\mathrm{iii}}(\mathcal{X},\mathcal{Y}) = h_1^{\mathrm{iii}}(\mathcal{Y}) - h_1^{\mathrm{iii}}(\mathcal{X}) \ , H_2^{\mathrm{iii}}(\mathcal{X},\mathcal{Y}) = h_2^{\mathrm{iii}}(\mathcal{Y}) - h_2^{\mathrm{iii}}(\mathcal{X}) \ , \end{cases}$$

where

$$\begin{aligned}
h_1^{\mathrm{i}}(\theta) &= \tfrac{d}{2\sin(\alpha)}(\tfrac{d^2}{2}\sin(\beta-\gamma+2\,\theta) - d(4\,b\sin(\alpha)\cos(\gamma-\theta)+ \\
&\quad d\theta\cos(\beta+\gamma)) + \tfrac{b^2}{2}\ln(-\tfrac{\sin(\beta+\theta)}{\cos(\gamma-\theta)})(2\,\sin(\beta+\gamma)- \\
&\quad \sin(2\,\alpha+\beta+\gamma)+\sin(2\,\alpha-\beta-\gamma)) + \sin^2(\alpha)(2\,b^2 \\
&\quad (\gamma-\theta)\cos(\beta+\gamma) - b^2\ln(\tan^2(\gamma-\theta)+1)\sin(\beta+\gamma))) \ , \\
h_2^{\mathrm{i}}(\theta) &= \tfrac{ad}{b\sin(\alpha)}(\tfrac{d^2\theta}{2}\cos(\beta) - \tfrac{d^2}{4}\sin(\beta+2\,\theta) + b^2\theta\cos(\beta)\sin^2(\alpha) \\
&\quad +2\,bd\sin(\alpha)\cos(\theta) - b^2\ln(\sin(\beta+\theta))\sin(\beta)\sin^2(\alpha)) \ , \\
h_1^{\mathrm{ii}}(\theta) &= \tfrac{bd}{4c\sin(\beta)}(d^2\sin(\gamma-2\,\theta) + 2\,d^2\theta\ \cos(\gamma) - 4\,c^2\sin^2(\beta) \\
&\quad (\ln(\sin(\theta))\sin(\gamma) - \theta\ \cos(\gamma)) + 8\,cd\sin(\beta)\cos(\gamma-\theta)) \ , \\
h_2^{\mathrm{ii}}(\theta) &= \tfrac{d}{4\sin(\beta)}(2\,d^2\theta\ \cos(\beta) - d^2\sin(\beta+2\,\theta) + 4\,c^2\sin^2(\beta) \\
&\quad (\ln(\sin(\theta))\sin(\beta) + \theta\ \cos(\beta)) + 8\,cd\cos(\beta+\theta)\sin(\beta)) \ , \\
h_1^{\mathrm{iii}}(\theta) &= \tfrac{ad}{4c\sin(\alpha)}(d^2\sin(\beta-2\,\theta) + 2\,d^2\theta\ \cos(\beta) + 8\,cd\sin(\alpha) \\
&\quad \cos(\theta) + 4\,c^2\sin^2(\alpha)(\theta\ \cos(\beta) + \sin(\beta)\ln(-\sin(\beta-\theta)))) \ , \\
h_2^{\mathrm{iii}}(\theta) &= \tfrac{2d}{\sin(\alpha)}(\tfrac{d^2}{8}\sin(\beta-\gamma+2\,\theta) - \tfrac{\theta}{4}\cos(\beta+\gamma)(2\,c^2\sin^2(\alpha)+d^2)- \\
&\quad cd\cos(\beta+\theta)\sin(\alpha) - \tfrac{c^2}{2}\ln(\sin(\gamma-\theta))\sin(\beta+\gamma)\sin^2(\alpha)) \ .
\end{aligned} \tag{4.20}$$

Therefore, the closed-form expression of the PDF of Ran2Ran NDD within an arbitrary triangle, i.e., (4.19), has been obtained. The corresponding CDF can also be obtained by integrating the PDF.

Although the obtained expression looks tedious, note that, for the network performance analysis, we do not use the symbolic expression of (4.19) directly but the numerical PDF result calculated promptly by (4.19) providing the necessary parameters (e.g., three vertexes of a triangle) of an arbitrary triangle. Simulations can also be utilized for obtaining NDDs. However, conducting simulations for each spe-

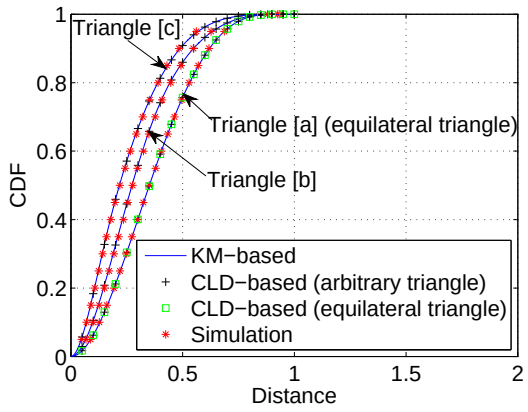


Figure 4.8: CDFs of the Ran2Ran NDDs within arbitrary triangles.

cific triangle is very time-consuming, and requires a large number of runs to obtain statistically significant results.

The obtained results are verified in comparison with simulation, which is conducted in Matlab as below (the following simulations are all conducted in a similar way):

(1) Generate a point uniformly at random within a triangle.

(2) Generate another point uniformly at random within the triangle.

(3) Compute the Euclidean distance between the two points and append the distance to a matrix.

(4) Repeat steps (1)–(3) $50,000$ times (the more repeats, the more accurate the result). Then using the Matlab function "ecdf" with the matrix as its input, we can obtain the empirical CDF of the Ran2Ran NDD within the triangle.

Three triangles are selected, all of which have the longest side length of 1, which can be scaled to any nonzero size as introduced before: [a] $\left(\frac{60\pi}{180}, \frac{60\pi}{180}, \frac{60\pi}{180}\right)$, [b] $\left(\frac{80\pi}{180}, \frac{70\pi}{180}, \frac{30\pi}{180}\right)$, and [c] $\left(\frac{130\pi}{180}, \frac{30\pi}{180}, \frac{20\pi}{180}\right)$. In [56], the CLD for an equilateral triangle was utilized to obtain the corresponding Ran2Ran NDD according to (4.3), and in our previous work [55], the CLD for an arbitrary triangle was utilized. Both are compared with our developed approach as well as simulation results, as shown in Figure 4.8.

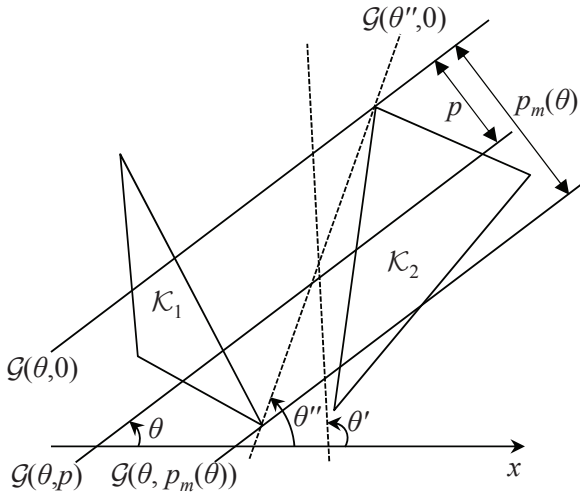


Figure 4.9: Ran2Ran NDD between two triangles $\mathcal{K}_1$ and $\mathcal{K}_2$.

4.4.2 Ran2Ran NDD between Two Triangles

Our approach can also be utilized to obtain the Ran2Ran NDD between two network geometries (e.g., two clusters in ad-hoc networks or two cells in cellular systems). As illustrated in Section 4.3.2 by using Figure 4.3a, the key is to calculate the length of the two chords (l_1 and l_3) respectively within the two geometries and the one segment (l_2) between the two geometries, which are produced by any line intersecting with the both two geometries. In this subsection, we show how to obtain the Ran2Ran NDD between any two triangles which are either disjoint or share a common vertex/side.

For illustration, Figure 4.9 shows an example of two arbitrary triangles $\mathcal{K}_1$ and $\mathcal{K}_2$ with area of S_1 and S_2, respectively. Define a line set $\mathcal{H} = \{\mathcal{G}|\mathcal{G}\cap\mathcal{K}_1 \neq \emptyset \text{ and } \mathcal{G}\cap\mathcal{K}_2 \neq \emptyset\}$. $\mathcal{H}(\theta)$ is a set of parallel lines with the same orientation θ ($0 \leq \theta \leq \pi$) and intersecting with the both triangles. Note that $\mathcal{H}(\theta)$ can be empty, if there is no line with orientation θ intersecting with the both triangles at all (e.g., $\mathcal{H}(\theta') = \emptyset$); otherwise, two support lines can always be found from $\mathcal{H}(\theta)$, such that all the other lines in $\mathcal{H}(\theta)$ are between them. So a line $\mathcal{G} \in \mathcal{H}(\theta)$ is determined by θ and its distance to, WLOG, any one of the two support lines, denoted as $\mathcal{G}(\theta, p)$, where $0 \leq p \leq p_m(\theta)$ and $p_m(\theta)$ is the distance between the two support lines, i.e., $\mathcal{G}(\theta, 0)$ and $\mathcal{G}(\theta, p_m(\theta))$. Note that it is possible that the two support lines are the same one for some orientations. For example, in Figure 4.9, $\mathcal{H}(\theta'')$ only contains one line, i.e., $\mathcal{G}(\theta'', 0)$ (so $p_m(\theta'') = 0$). With the above analysis and based on (4.6) and (4.8), we can design an algorithmic approach to obtain the Ran2Ran NDD between any two

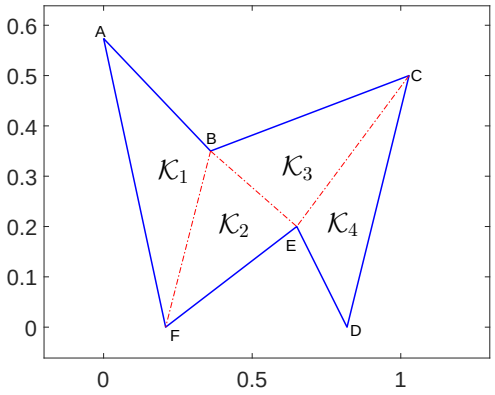


Figure 4.10: An example of four triangles, $\mathcal{K}_1(\triangle ABF)$, $\mathcal{K}_2(\triangle BEF)$, $\mathcal{K}_3(\triangle BCE)$, and $\mathcal{K}_4(\triangle CDE)$.

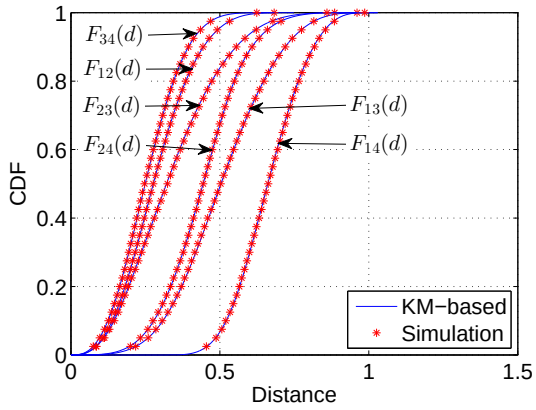


Figure 4.11: CDFs of the Ran2Ran NDDs between any two triangles shown in Figure 4.10.

triangles which are either disjoint or share a common vertex/side.

Figure 4.10 shows an example with four triangles, $\mathcal{K}_1(\triangle ABF)$, $\mathcal{K}_2(\triangle BEF)$, $\mathcal{K}_3(\triangle BCE)$, and $\mathcal{K}_4(\triangle CDE)$. With the developed approach, the CDFs of the Ran2Ran NDDs between any two out of the four triangles can be obtained, which have a close match with the simulation results, as shown in Figure 4.11.

4.4.3 Ran2Ran NDD Associated with Arbitrary Polygons

With the triangle-NDDs obtained above, the Ran2Ran NDDs associated with arbitrary polygons can be obtained through D&R, since any polygon can be triangulated. Therefore, the Ran2Ran NDD-based performance metrics of wireless networks

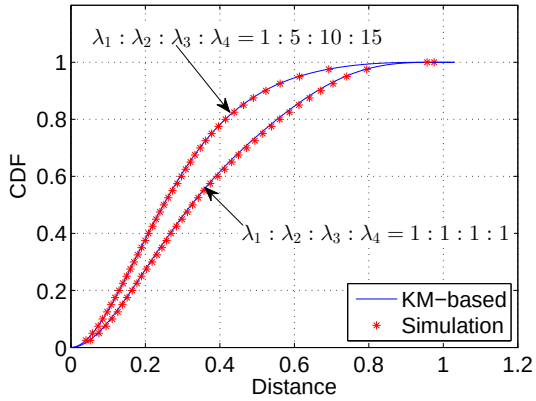


Figure 4.12: CDFs of the Ran2Ran NDD within the polygon shown in Figure 4.10.

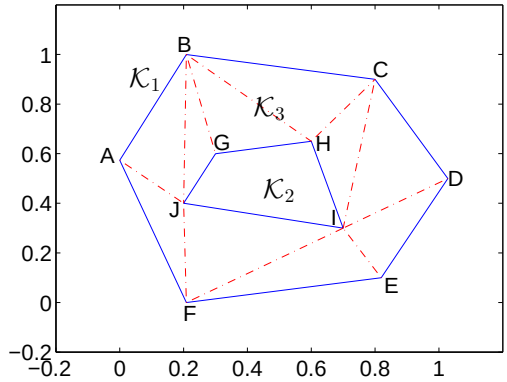


Figure 4.13: An example of tiered polygons, where a polygon $\mathcal{K}_1$ contains another polygon $\mathcal{K}_2$, with the ring area labeled as $\mathcal{K}_3$.

associated with arbitrary polygons can be quantified properly.

Take the irregular polygon $ABCDEF$ with the triangulation shown as in Figure 4.10 for example. Assuming the node density ratio among $\mathcal{K}_1$, $\mathcal{K}_2$, $\mathcal{K}_3$, and $\mathcal{K}_4$ is $\lambda_1 : \lambda_2 : \lambda_3 : \lambda_4$ (λ_1, λ_2, λ_3, and λ_4 are not zero at the same time), through D&R, the CDF of the Ran2Ran NDD within the polygon is given by a probabilistic sum,

$$F(d) = \sum_{i=1}^{4} \sum_{j=1}^{4} \frac{S_i \lambda_i S_j \lambda_j}{(\sum_{k=1}^{4} S_k \lambda_k)^2} F_{ij}(d) \ ,$$

where S_x is the area of triangle $\mathcal{K}_x$, and $F_{ij}(d)$ or $F_{ji}(d)$ is the CDF of the Ran2Ran NDD within triangle $\mathcal{K}_i$ if $i = j$, or between two triangles $\mathcal{K}_i$ and $\mathcal{K}_j$ if $i \neq j$, which have been obtained above. The results compared with simulation are shown in

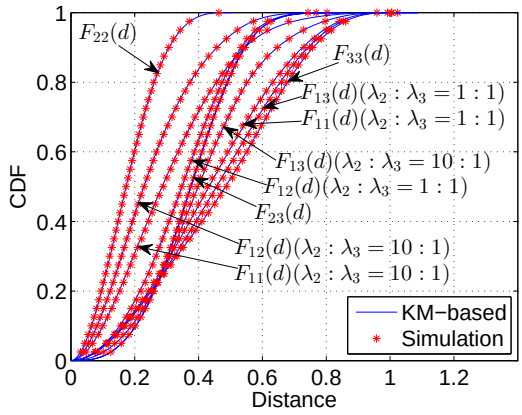


Figure 4.14: CDFs of the Ran2Ran NDDs associated with the tiered polygons shown in Figure 4.13.

Figure 4.12.

Figure 4.13 shows an example of tiered polygons, where a polygon $\mathcal{K}_1$ contains another polygon $\mathcal{K}_2$, with the ring area (the area of $\mathcal{K}_1$ minus the area of $\mathcal{K}_2$) labeled as $\mathcal{K}_3$. $\mathcal{K}_3$ can be triangulated as shown in the figure. With the triangle-NDDs obtained based on the above approach, the CDFs of the Ran2Ran NDDs within $\mathcal{K}_1$, $\mathcal{K}_2$, and $\mathcal{K}_3$, i.e., $F_{11}(d)$, $F_{22}(d)$, and $F_{33}(d)$, can all be obtained. Then the CDF of the Ran2Ran NDD between $\mathcal{K}_2$ and $\mathcal{K}_3$, i.e., $F_{23}(d)$, can be obtained according to (4.11), through which the CDF of the Ran2Ran NDDs between $\mathcal{K}_1$ and $\mathcal{K}_2$ ($F_{12}(d)$) and that between $\mathcal{K}_1$ and $\mathcal{K}_3$ ($F_{13}(d)$) can also be obtained. Given that the node density ratio between $\mathcal{K}_2$ and $\mathcal{K}_3$ is $\lambda_2 : \lambda_3 = 1 : 1$ or $10 : 1$, Figure 4.14 shows the corresponding results in comparison with the simulation results.

4.5 Conclusions

In this chapter, we have developed a novel, systematic and algorithmic approach to obtain the Ran2Ran NDDs associated with arbitrarily-shaped networks. The proposed approach fills the gap in the literature such that the network shape can be approximated more accurately based on the realistic scenarios, and the NDD-based analysis associated with concave, disjoint, and tiered networks can be conducted. We also surveyed the state-of-the-art approaches to the Ref2Ran NDDs with arbitrary shapes and nonuniform densities. The implementation of the proposed approach in Matlab can be found in [154].

Chapter 5

Physical Interference Modeling and Analysis

5.1 Overview

In this chapter, we first categorize the existing NDD applications in the current literature into different levels in Section 5.2. Then we propose a physical interference model framework based on NDD to analyze the cumulative interference at a receiver and link outage probability for an LSN running the PDC protocol in Section 5.3. Not just for linear networks, the framework is further utilized to analyze 2D networks, as shown in Section 5.4.

5.2 Categorizing NDD Applications in WANETs

NDDs (including both Ref2Ran and Ran2Ran NDDs) can be utilized to characterize most of the performance metrics in finite WANETs due to their nonlinear relationships with the distances among nodes. In this section, we categorize the existing applications in the current literature into different levels, including graph, transceiver, link, path, and network levels. We also show the efficacy of the proposed approach on some selected performance metrics with arbitrary shapes/densities, in comparison with previous approaches/approximations such as using average density, ignoring border effect (e.g., in PPP), and so on.

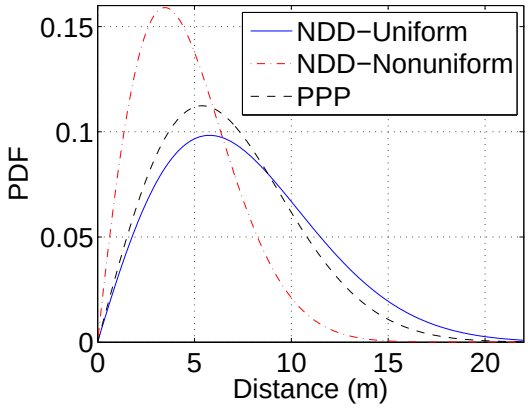


Figure 5.1: Nearest neighbor distance distribution.

5.2.1 Graph Level

There are several representative performance metrics at the graph level, such as kth Nearest Neighbor (k–NN) distance, node degree (just k–NN below a certain threshold), etc. Especially, 1–NN and $(n-1)$–NN (n is the total number of nodes in the network) correspond to the nearest and farthest neighbor distances, respectively, which are useful for routing protocol design in ad hoc networks [48, 53]. For example, in a sparse network where the network size is much larger than the communication range of the nodes, a nearest-neighbor routing is beneficial to reducing energy consumption and increasing network throughput. On the other hand, in a small dense network, choosing the farthest node for packet relay, the routing overhead can be alleviated by reducing the number of transmissions. In addition, the nearest neighbor distance distribution was also utilized in [28, 29] to evaluate the nearest-job-next service discipline for mobile collectors or chargers (known as mobile elements).

Suppose there are n nodes uniformly distributed at random in a network area. For a node i (either a random or reference node), the distances from the other $n-1$ nodes to node i are ordered as $d_1 \leq d_2 \leq \cdots \leq d_{n-1}$. Let Δ_k denote the random variable which represents the k–NN distance to node i. The PDF of Δ_k according to order statistic is

$$f_{\Delta_k}(d) = \frac{(n-1)!}{(k-1)!(n-1-k)!}[F(d)]^{k-1}[1-F(d)]^{n-1-k}f(d) \ , \tag{5.1}$$

where $F(d)$ and $f(d)$ are the CDF and PDF of any NDD obtained in Section 4.4, respectively.

Suppose there are $n = 10$ nodes randomly distributed in $\mathcal{P}_3$ shown in Figure 4.1c. For $\lambda_1 : \lambda_2 = 1 : 1$ (i.e., uniform distribution) and $10 : 1$ (nonuniform distribution), Figure 5.1 shows the corresponding Ran2Ran nearest neighbor distance distributions, compared with the result obtained based on the PPP model. Since PPP ignores the network border effect and due to the different node density ratios, there exist gaps among the comparisons shown in Figure 5.1. Also in the nonuniform case where $\mathcal{P}_1$ has a higher density, surrounded by $\mathcal{P}_2$ with a lower density, nodes are more likely closer to each other in $\mathcal{P}_1$, with nearer nearest-neighbors than the uniform case, as shown in Figure 5.1.

5.2.2 Transceiver Level

The performance metrics at the transceiver level include path loss [23], received signal strength for a given transmission power, transmission energy consumption to ensure a certain received power [27, 53, 155], etc.

Path Loss and Signal Strength

Let us assume a general path-loss model, where the path loss of the transmission power at distance d is

$$L(d) = \beta d_0^{\alpha} d^{-\alpha} \ , \tag{5.2}$$

where β is a path-loss constant determined by the hardware features of transceivers, d_0 is a given reference distance, and α is the path-loss exponent. As a result, the received signal strength at distance d is just

$$P_r(d) = L(d) P_t \ , \tag{5.3}$$

where P_t is the transmission power. Therefore, given a distance distribution, the distributions of path loss and received signal strength can also be obtained by using the change-of-variable technique. The model can be readily extended to include the shadowing and fading effects of wireless channels. For example, log-normal shadowing and Rayleigh fading can be considered. For the Rayleigh fading channel, we have the PDF of the channel power gain as

$$f_X(x|d) = \frac{1}{P_r(d)} \mathrm{e}^{\frac{-x}{P_r(d)}} \ . \tag{5.4}$$

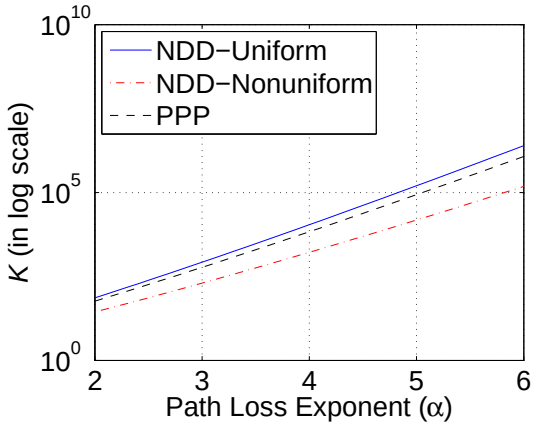


Figure 5.2: Nearest neighbor energy consumption (K vs. α).

Then the PDF of the signal strength at the receiver is

$$f_X(x) = \int_{d_{\min}}^{d_{\max}} f_X(x|d) f(d) \mathrm{d}d \ , \tag{5.5}$$

where $f(d)$ is the PDF of any NDD, and $d_{\min}$ and $d_{\max}$ are the minimum and maximum distances between the transmitter and receiver, respectively. In addition, log-normal shadowing and Rayleigh fading can also be modeled as independent random variables that are not related to inter-node distances. As shown in [156], the shadowing effect follows a log-normal distribution with standard deviation σ (typically between zero and eight dB), and Rayleigh fading follows an exponential distribution of mean 1. Therefore, the extension of the path-loss model along with shadowing and fading is the multiplication of independent random variables and can still be analyzed based on the NDD-based model.

Transmission Energy Consumption

The energy consumed by a radio transmitter is proportional to the αth power of the distance to the receiver. In a simplistic model with wide applicability [27, 155], the average one-hop energy consumption of the radio transmitter can be formulated as

$$E_{\mathrm{Tx}} = \epsilon \int d^{\alpha} f(d) \mathrm{d}d = \epsilon K \ , \tag{5.6}$$

where ϵ is a constant related to the environment, $f(d)$ is the PDF of any relevant NDD, and K can be viewed as the normalized average bit-energy consumption. Based

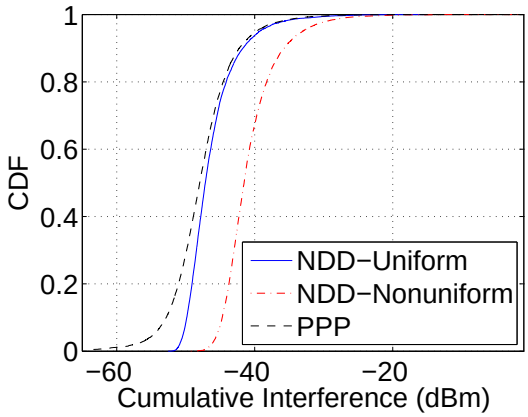


Figure 5.3: Cumulative interference at R shown in Figure 4.1c.

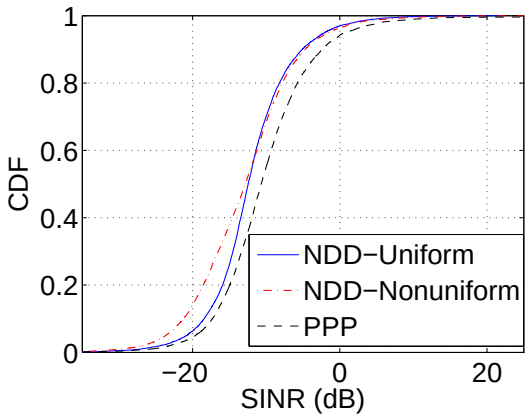


Figure 5.4: SINR achieved at R shown in Figure 4.1c.

on the obtained NDDs shown in Figure 5.1, Figure 5.2 shows the variations of K as α increases. Due to the nonlinear effect of the path loss exponent, even a small difference in distance distributions can lead to a big difference in energy consumption. Again, PPP-based model differs from the reality due to the ignored border effect, and nonuniform node distribution also has a great effect.

5.2.3 Link Level

The interference [21, 22], SINR [22], outage (just SINR below a certain threshold) [23], link capacity [22], etc., achieved at either a random or fixed receiver are link-level performance metrics. Assuming that all the transmitters in the network have the same transmission power P_t, the cumulative interference at the receiver

from all its interfering nodes is

$$I = P_t \sum_i L(d_i) \ , \tag{5.7}$$

where $L(d_i)$ is given in (5.2), and d_i is the distance from the receiver to the ith interfering node. So the SINR achieved at the receiver is

$$SINR = \frac{P_t L(d)}{N_o W + I} \ , \tag{5.8}$$

where d is the distance from the receiver to its transmitter, W is the communication bandwidth, and N_0 is the one-sided spectral density of additive white Gaussian noise. Given a modulation and coding scheme, outage probability represents the chance that the SINR achieved at a receiver is no larger than a specified threshold so that the reception is considered unsuccessful. Therefore, the CDF of the received SINR is significant to determine the link outage probability. Meanwhile, according to Shannon's theory, the capacity of the link between the transmitter and receiver is

$$C = W \log_2(1 + SINR) \ . \tag{5.9}$$

Since I, $SINR$, and C are all functions of distance, given the corresponding NDD, their distributions can also be obtained, which are significant for statistically analyzing the performance of ad hoc networks. For the network shown in Figure 4.1c with $n = 10$ nodes randomly distributed in $\mathcal{P}_3$, Figure 5.3 and Figure 5.4 show the cumulative interference from the other 9 interferers and SINR at R, respectively, for both $\lambda_1 : \lambda_2 = 1 : 1$ and $10 : 1$, in comparison with the result obtained based on the PPP model ($P_t = 2$ mWatt, $L(d) = -38 - 20\text{lg}(d)$ (dB), $W = 5$ MHz, and $N_0 = -174$ dBm/Hz). Higher density in $\mathcal{P}_1$, surrounding R, thus causes a higher interference and yields a lower SINR at R, as shown in the figures.

5.2.4 Path Level

The metrics at the link level shown above are utilized to investigate the performance of single-hop communications (i.e., via a direct link). For analyzing multi-hop transmissions at the path level, NDD can still be utilized. For example, hop distance is crucial to the route discovery delay, the reliability of message delivery, and the minimization of multi-hop energy consumption. The authors in [25] investigated the

distribution of the minimum hop distance H between a random source and destination pair based on NDD. The closed-form expressions for the probability that two nodes can communicate within $H = 1$ hop or $H = 2$ hops were derived. Analytical bounds were provided for the paths with $H > 2$ hops. In [157], the NDD between a fixed source and destination pair with a single relay uniformly distributed at random in between is utilized to obtain the distribution of the capacity of the two-hop relay communication.

5.2.5 Network Level

The analysis on the network capacity belongs to this level. Network capacity can be investigated from the perspective of either concurrent links or flows. For example, in a clustered ad hoc network, there are concurrent single-hop communications between cluster members and their heads in several clusters, where the network capacity can be obtained based on the link capacity. On the other hand, in an ad hoc mesh network, there might be several multi-hop communications (referred to as flows) happening concurrently. The network transport capacity in this case can be investigated based on the capacity studied at the path level.

As shown above, ignoring border effect or using average density often in PPP and existing work skews results greatly, which highlights the need for NDDs to analyze performance metrics accurately in finite ad hoc networks.

5.3 Analyzing LSNs Running PDC

For the PDC protocol proposed in Chapter 2 and modeled in Chapter 3, the sleep factor ξ in (2.2) is set to at least two based on the assumption that interference is about twice the range of transmission, which is to ensure that the communication between two adjacent grades will not be interfered with by their upper or lower grades. It defines location-based conditions, i.e., transmission and interference ranges, for successful communications between a pair of nodes, and does not take into account the cumulative interference from the transmitters which are out of the interference range of a receiver. In this section, we propose a physical interference model framework based on NDD to analyze the cumulative interference at a receiver and link outage probability for an LSN running the PDC protocol.

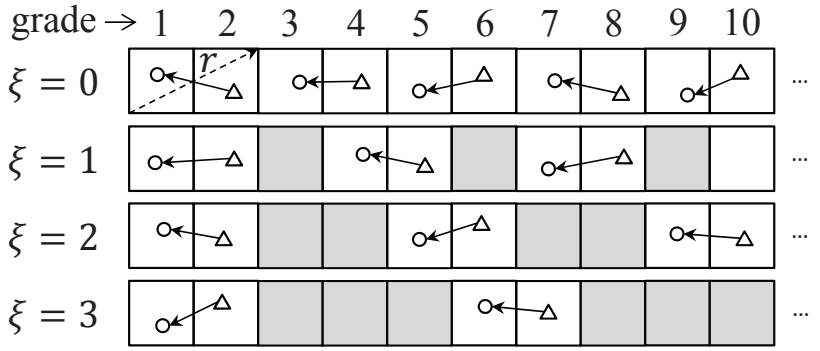


Figure 5.5: System Model with four LSNs given four different sleep factor ξ. $\square$: the grades in state R or T, $\blacksquare$: the grades in state S, $\triangle$: transmitter, $\circ$: receiver.

5.3.1 System Model

We consider a linear sensor network running the PDC protocol. With PDC, the network is divided into N_G ($N_G \in \mathbb{N}$ and $N_G \geq 2$) grades based on the locations of the monitored areas. Sensor nodes are independently and uniformly distributed within the area of each grade. A node can only send data to one of its adjacent lower-grade nodes, with a transmission power of P_t. Meanwhile, two nodes can communicate with each other only when their distance is within a predefined range $[d_{\min}, d_{\max}]$. We will carry out interference analysis for the communication between grades one and two. The same way can be applied to the communications between other two adjacent grades. Assume at the moment, the nodes in grade $1 + i(\xi + 2)$ are in state R and those in grade $2 + i(\xi + 2)$ ($i \in \mathbb{N}^0$) are in state T, and there is one communication successfully established between the two adjacent grades. Therefore, the receiver in grade one will be interfered with by the transmitter in grade $2 + i(\xi + 2)$ ($i \in \mathbb{N}^+$). So, totally there are $N_I = \lfloor \frac{N_G - 2}{\xi + 2} \rfloor$ interfering transmitters. Figure 5.5 shows four LSNs given four different ξ (0, 1, 2, and 3) for illustration. Note that the grade region in the figure is shown as a square just for the ease of presentation, and it can be in arbitrary shape.

Considering that wireless signal power is attenuated with distance, a general path-loss model is applied and according to (5.8) and (5.9), the received SINR at the receiver in grade one and link capacity, respectively, are

$$SINR_1 = \frac{K P_t {d_0}^{\alpha} {d_{12}}^{-\alpha}}{N_0 W + I_1} \ , \tag{5.10}$$

and

$$C_{12} = W \log_2(1 + SINR_1) \ , \tag{5.11}$$

where K is the antenna- and processing gain-related parameter, d_0 is the reference distance, α is the path-loss exponent, d_{12} is the distance between the receiver and transmitter in grades one and two, respectively, W is the communication bandwidth, N_0 is the one-sided spectral density of additive white Gaussian noise, and I_1 is the cumulative interference at the receiver in grade one from all the N_I interfering transmitters and we have

$$I_1 = K P_t {d_0}^{\alpha} \sum_{i}^{N_I} {d_{1i}}^{-\alpha} \ , \tag{5.12}$$

where d_{1i} is the distance between the receiver and the ith interfering transmitter.

From (5.10)–(5.12), it is obvious that the distances, i.e., d_{12} and d_{1i}, play significant roles. Since signal and interference are non-linear functions of inter-node distances, not only their mean values but also their distributions are needed to obtain the statistics of SINR using (5.10), all based on the distributions of the distances. With the obtained SINR statistics, the performance metrics of the network, such as outage probability and link capacity, can be quantified effectively.

5.3.2 Approaches to Distance, Interference, SINR, and Link Capacity Distributions

Distance Distributions

With the approach to Ran2Ran NDD proposed in Section 4.3.2 for disjoint networks, the required distance distributions between two grade regions can be obtained.

Interference Distribution

With the obtained distance distributions, the SINR distributions can be obtained according to (5.10)–(5.12). To this end, we first obtain the distribution of $d^{-\alpha}$ using the change-of-variable technique, based on which the distribution of the signal or interference from one source achieved at a receiver can be obtained. Specifically, let D be an Random Variable (RV) of a distance with PDF $f_D(d)$ defined over $d_1 \leq d \leq d_2$. A new RV, $D^{-\alpha}$, is introduced. Let $I = D^{-\alpha} = u(D)$, and $D = I^{-\frac{1}{\alpha}} = v(I)$. Then

the CDF $F_I(i)$ defined over $u(d_2) \leq i \leq u(d_1)$ is

$$F_I(i) = \Pr(I \leq i) = \Pr(u(D) \leq i) = \Pr(D \geq v(i)) = 1 - \int_{d_1}^{v(i)} f_D(d)\ \mathrm{d}d\ . \quad (5.13)$$

Thus, the PDF is

$$f_I(i) = F_I'(i) = f_D(v(i)) \cdot |v'(i)|\ . \quad (5.14)$$

To obtain the distribution of cumulative interference from multiple sources, e.g., N_I interfering transmitters, the convolution operator is used for the PDF of the sum of independent RVs, as shown below in general,

$$I(x) = \psi^{(N)}(x)\ , \quad (5.15)$$

where $I(x)$ represents the distribution of the total interference from all N_I interfering transmitters, and $\psi^{(z)}(\cdot)$ is the z-fold convolution of $\psi(\cdot)$.

SINR Distribution

In this subsection, we show in detail the analysis of the SINR distribution at the receiver in grade one. Specifically, let the SINR obtained in (5.10) in dB be

$$S = 10\lg(SINR_1)\ . \quad (5.16)$$

Obviously, S depends on the distance from the receiver to its transmitter (denoted as D, within the communication range $[d_{\min}, d_{\max}]$) and the cumulative interference from all interfering transmitters (denoted as I, $I \in [I_{\min}, I_{\max}]$). Let $f_D(d)$ and $f_I(i)$ denote the PDFs of D and I, respectively, which can be obtained following the approaches explained above.

Let $\mathtt{f}_S(D, I)$ represent the received SINR at the receiver given that the distance between the receiver and the transmitter is D and the cumulative interference from all interfering transmitters to the receiver is I, $\mathtt{f}_D(S, I)$ be the distance given the received SINR S and interference I, and $\mathtt{f}_I(S, D)$ be the interference given the received SINR S and distance D. Thus, $\mathtt{f}_S(d_{\max}, I_{\max}) \leq S \leq \mathtt{f}_S(d_{\min}, I_{\min})$. Then the CDF of the received SINR in dB at the receiver is:

if $\mathtt{f}_S(d_{\max}, I_{\min}) \leq \mathtt{f}_S(d_{\min}, I_{\max})$, then $\Pr(S \leq s) =$

$$\begin{cases} \int_{\mathtt{f}_I(s,d_{\max})}^{I_{\max}} \int_{\mathtt{f}_D(s,i)}^{d_{\max}} f_D(d) f_I(i) \ \mathrm{d}d\mathrm{d}i, \\ \qquad \text{if } \ \mathtt{f}_S(d_{\max}, I_{\max}) \leq s \leq \mathtt{f}_S(d_{\max}, I_{\min}), \\ \int_{I_{\min}}^{I_{\max}} \int_{\mathtt{f}_D(s,i)}^{d_{\max}} f_D(d) f_I(i) \ \mathrm{d}d\mathrm{d}i, \\ \qquad \text{if } \ \mathtt{f}_S(d_{\max}, I_{\min}) \leq s \leq \mathtt{f}_S(d_{\min}, I_{\max}), \\ 1 - \int_{I_{\min}}^{\mathtt{f}_I(s,d_{\min})} \int_{d_{\min}}^{\mathtt{f}_D(s,i)} f_D(d) f_I(i) \ \mathrm{d}d\mathrm{d}i, \\ \qquad \text{if } \ \mathtt{f}_S(d_{\min}, I_{\max}) \leq s \leq \mathtt{f}_S(d_{\min}, I_{\min}). \end{cases}, \tag{5.17}$$

otherwise, $\Pr(S \leq s) =$

$$\begin{cases} \int_{\mathtt{f}_I(s,d_{\max})}^{I_{\max}} \int_{\mathtt{f}_D(s,i)}^{d_{\max}} f_D(d) f_I(i) \ \mathrm{d}d\mathrm{d}i, \\ \qquad \text{if } \ \mathtt{f}_S(d_{\max}, I_{\max}) \leq s \leq \mathtt{f}_S(d_{\min}, I_{\max}), \\ \int_{d_{\min}}^{d_{\max}} \int_{\mathtt{f}_I(s,d)}^{I_{\max}} f_D(d) f_I(i) \ \mathrm{d}i\mathrm{d}d, \\ \qquad \text{if } \ \mathtt{f}_S(d_{\min}, I_{\max}) \leq s \leq \mathtt{f}_S(d_{\max}, I_{\min}), \\ 1 - \int_{I_{\min}}^{\mathtt{f}_I(s,d_{\min})} \int_{d_{\min}}^{\mathtt{f}_D(s,i)} f_D(d) f_I(i) \ \mathrm{d}d\mathrm{d}i, \\ \qquad \text{if } \ \mathtt{f}_S(d_{\max}, I_{\min}) \leq s \leq \mathtt{f}_S(d_{\min}, I_{\min}). \end{cases}. \tag{5.18}$$

With the developed numerical algorithm based on (5.17) and (5.18), we can obtain the corresponding numerical results promptly and accurately according to different network settings and parameters.

Link Capacity Distribution

With the SINR distributions obtained above, the link capacity distributions can be obtained with (5.11) by using the change-of-variable technique. Specifically, let S be an RV of SINR with PDF $f_S(s)$ defined over $s_1 \leq s \leq s_2$. A new RV, $C = W \log_2(1+S)$, is introduced. Let $C = u(S)$, and $S = 2^{\frac{C}{W}} - 1 = v(C)$. Then the CDF $F_C(c)$ and PDF $f_C(c)$ defined over $u(s_1) \leq c \leq u(s_2)$ are respectively shown as

$$F_C(c) = \Pr(C \leq c) = \Pr(u(S) \leq c) = \Pr(S \leq v(c)) = \int_{s_1}^{v(c)} f_S(s) \ \mathrm{d}s \ , \tag{5.19}$$

$$f_C(c) = F_C'(c) = f_S(v(c)) \cdot |v'(c)| \ . \tag{5.20}$$

5.3.3 Performance Evaluation

In this subsection, the obtained analytical results from the framework are validated in comparison with simulations in terms of cumulative interference and outage probability. It can provide insights into tuning the protocol parameters, such as ξ, given the outage probability of sensor nodes. There is an obvious tradeoff in setting ξ in PDC. A smaller ξ may lead to more concurrent transmissions, while the cumulative interference to the receiver of each communicating node pair increases as well, which more likely results in unsuccessful communications. The proposed physical interference modeling and analysis can help select a proper ξ.

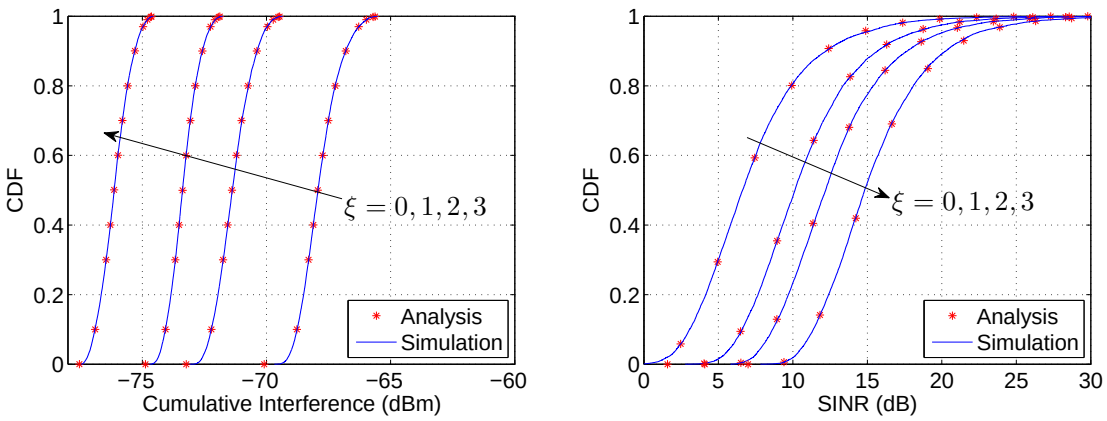
Considering the network model shown in Section 5.3.1, for the ease of presentation and WLOG, each grade has an identical square area, with the side length of $\sqrt{\frac{r^2}{5}}$, where r is the length of the diagonal of the rectangle formed by two adjacent squares (so, $d_{\max} = r$). Note that the proposed framework does not impose any constraints on the shapes of the grade region, since the proposed approach to distance distributions in Chapter 4 can handle arbitrarily-shaped networks. $N_G = 10$, $d_{\max} = r = 30$ m, $d_{\min} = 1$ m, $P_t = 1$ mWatt, $W = 250$ Kbps, $N_0 = -174$ dBm/Hz. A log-distance path-loss model is applied, with the path-loss ratio set as $L(d) = -38 - 20\lg(d)$ (dB), so the path-loss exponent $\alpha = 2$. The numerical results are obtained based on our analytical framework directly; on the other hand, the simulations are conducted in Matlab as follows,

(1) Generate one sensor node uniformly at random for each grade region.

(2) Compute the received interference and SINR at the grade-1 node with the above network parameters and append the computed results to the interference and SINR matrixes, respectively.

(3) Repeat steps (1)–(2) $50,000$ times (the more repeats, the more accurate the result). Then using the Matlab function "ecdf" with each matrix as its input, we can obtain corresponding empirical CDFs.

Given a modulation and coding scheme, outage probability represents the chance that the SINR achieved at a receiver is no larger than a specified threshold and the reception is considered unsuccessful. Therefore, the CDF of the received SINR (as analyzed in Section 5.3.2) is significant to determine the outage probability of a communication link.



(a) Interference Distribution (b) SINR Distribution

Figure 5.6: Distributions of the interference and SINR at the grade-1 receiver with $\xi = 0, 1, 2$, and 3.

With ξ varying from zero to three, the CDFs of the cumulative interference and SINR received at the grade-1 receiver are shown in Figure 5.6a and Figure 5.6b, respectively. Both figures show a good match between the analytical and simulation results. With a smaller ξ, there are either more or closer interfering transmitters, imposing higher interference on the grade-1 receiver. So for the cumulative interference distribution shown in Figure 5.6a, a smaller ξ corresponds to the CDF curves located on the right-hand side; while for the SINR distribution shown in Figure 5.6b, it corresponds to the CDF curves located on the left-hand side.

The communications between other grades can be investigated in the same way. Given a bounded outage probability of the receiving nodes, a proper ξ can be selected for PDC to ensure a successful communication with high probability between each two grades respectively in state `R` and `T`, with as many concurrent communications as possible.

5.4 Analyzing Underlaying Ad Hoc D2D Communications

Not only for linear networks, but NDD can also be utilized to analyze 2D networks. In this section, the physical interference model framework is further applied to analyze the ad hoc D2D communications underlaying a cellular network to investigate the cumulative interference and outage probabilities for both cellular and D2D

communications.

5.4.1 Background

There have been great research efforts in obtaining the performance metrics for D2D communications in cellular networks based on interference analysis. For example, Cheng et al. [63] studied the uplink capacity gain when a D2D link is enabled in an FDD CDMA-based cellular network. In [64], an Interference-Limited Area (ILA) control scheme for an uplink reusing scenario was proposed to manage the interference from the CUE to a D2D transmission when multiple antennas are used by the BS. By analyzing the coverage of the ILA for one D2D pair, a lower bound of the ergodic capacity was derived. After that, an extended approach was proposed in [65] to take into consideration a downlink reusing scenario. Overall, these works assumed that there is only one D2D pair in a cell to avoid harmful interference to the cellular transmissions in the cell, and thus their results cannot be directly applied to the scenario with multiple concurrent D2D pairs.

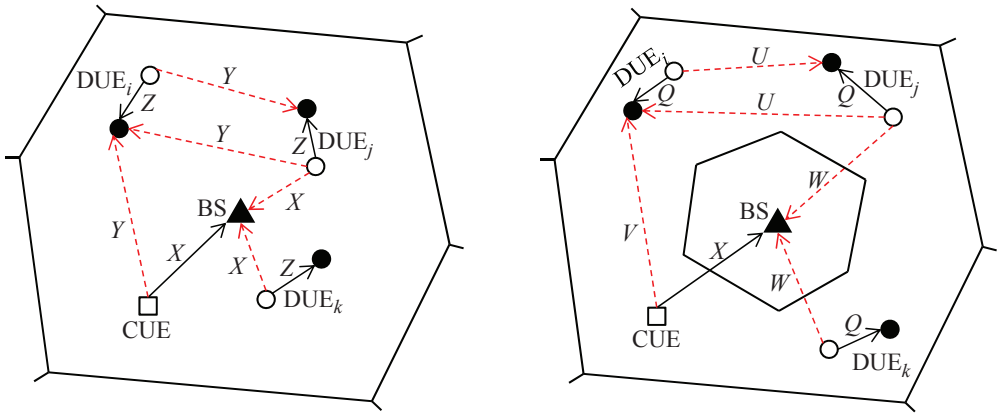
Recent works begin to consider the interference from multiple D2D pairs accumulated at the BS in a cell. For example, Ma et al. [68] studied the effect of the Successive Interference Cancellation (SIC) technique, aiming to reduce/control interference and improve network capacity, in a D2D-enabled, multi-cell cellular network. By using the tools from stochastic geometry, the successful transmission probabilities for both the cellular uplinks and D2D links with SIC were derived. In [69], an uplink reusing scenario in a multi-cell, OFDMA-based cellular network was considered. Both the Guard Regions (GRs) approximated as circles centered at each BS and an open-loop fraction power control scheme were adopted to mitigate the D2D-to-cellular interference. Also with stochastic geometry, the outage probabilities of both CUEs and DUEs and the area spectral efficiency of the whole hybrid network were explored. To simplify the analysis, the model assumed that the D2D transmitter and its intended receiver in a pair have a constant distance, which might be impractical due to the randomness of DUE locations. In summary, these works considered a large-scale cellular system with multiple cells working on the same frequency band (i.e., the frequency reuse factor is one). With the tools from stochastic geometry, BSs were assumed to be deployed randomly following a Poisson point process, whereas in reality BSs are deployed according to cell planning and their locations are not totally random. Especially, in a scenario with a higher frequency reuse factor, e.g., seven, where it needs

to investigate the performance metrics of a single cell, or for a sector-partitioned cell where the BS is equipped with multiple directional antennas [67], their results are not applicable directly.

As an another example with stochastic geometry, Syu et al. [158] analyzed the spatial constraints of DUEs in a single-cell, uplink reusing scenario. Depending on whether there is a GR for the BS and the BS serving zone is infinite or not, four kinds of network models were discussed. Both the cell and GR were modeled as circles. For the cumulative interference to the BS, only the D2D transmitters located at a fixed distance from the BS were considered. To analyze the success probability of D2D transmissions, the model assumed that the transmission power of the CUE is no larger than that of DUE, which might be impractical since the DUE transmission power is usually much smaller than the CUE's. In addition, for the finite coverage case, the model only worked out the success probability of the CUE transmission.

For the D2D throughput bound analysis, a geometrical-based method was proposed in [66] considering an uplink reusing scenario. Three guard distances from a DUE, respectively to the BS, to the CUE, and to the other communicating DUEs were obtained by considering the SINR requirements for both cellular and D2D transmissions. Then the maximum number of concurrent D2D pairs was approximated, and further the D2D throughput bound was derived. The bound analysis was then extended in [67] to analyze the interference accumulated at the BS and receiving DUEs.

Recently, there are increasing studies on the utilization of distance distributions for performance analysis in wireless networks. In [146], the total interference received at a BS in a cell from the transmissions in its neighbor cells was obtained based on the distance distributions associated with regular hexagons. A general framework based on the distance distributions associated with arbitrary shapes was proposed in [145] to analyze the outage probability in finite wireless networks. For tiered cellular networks, our recent work [20] utilized distance distributions to analyze the outage probability for the macro and femto BSs in arbitrarily-shaped cells. Overall, they are all based on the distribution of the distance between a given reference point (e.g., the BS or a given reference receiver) and a random point. For analyzing the underlaying D2D communications in cellular networks, however, the above approaches/results are not applicable, since all DUEs are randomly distributed and thus the distribution of the distance between two random points has to be involved.



(a) No GR for the BS (b) With GR for the BS

Figure 5.7: System model consisting of several D2D pairs (shown as circles) underlaying a cell (shown as an irregular polygon) with one CUE (shown as a square, which can be anywhere in the cell region) in an uplink reusing scenario, where the solid black arrow lines show the transmission between two DUEs or between the CUE and BS, and the dashed red arrow lines show the interference at a receiver from an unintended transmitter. Note that not all of the unintended transmitters' interference is shown for a receiver.

5.4.2 System Model

The system model considered in this chapter is shown in Figure 5.7, where the UEs are independently and uniformly distributed within a polygonal cell (the non-uniform UE distribution can also be considered as discussed in Section 5.4.4). A higher frequency reuse factor is considered, so that the neighbour cells are in orthogonal channels. To ensure the generality and applicability of our approach, the cell is shown with an arbitrary polygonal shape. CUEs communicate with the BS in either uplink or downlink mode with a transmission power of P_C. In the uplink mode, CUEs transmit to the BS, while the direction is reversed in the downlink mode. On the other hand, DUEs exchange data directly by utilizing either the uplink or downlink cellular resources with a transmission power of P_D. Since uplink resources are more likely to be shared than downlink resources due to the asymmetric uplink and downlink service loads, under-utilization of uplink spectrum, and stronger abilities at BSs to process interference than at CUEs [70–72], this chapter focuses on an uplink reusing scenario, in which one CUE and N concurrent D2D pairs are transmitting simultaneously reusing the uplink resource of the CUE. Note that the active CUE location is random at a given time. For multiple CUEs in orthogonal channels, the proposed framework applies to each of them. For a downlink reusing scenario, our

approach still applies, as explained in Section 5.4.4. A D2D connection will be established only when the distance between two DUEs is within a predefined range $[d_{\min}^{D2D}, d_{\max}^{D2D}]$. Depending on whether or not there is a GR set for the BS to guarantee the successful transmission from the CUE to the BS, two scenarios are considered: 1) no GR set for the BS, as shown in Figure 5.7a; 2) a GR set for the BS, as shown in Figure 5.7b. For the latter case, the D2D communications only happen outside the GR, whereas the active CUE can be anywhere within the cell. In this chapter, we will conduct a physical interference model-based performance analysis, which focuses on the cumulative interference at a specific receiver.

Similar to the analysis shown in Section 5.3.1, the received SINR at the BS from the CUE is

$$SINR_B = \frac{KP_C {d_0}^{\alpha} {d_{CUE,BS}}^{-\alpha}}{N_0 W + I_B} , \tag{5.21}$$

and the corresponding link capacity is

$$C_B = W \log_2(1 + SINR_B) , \tag{5.22}$$

where K is the antenna- and processing gain-related parameter, d_0 is the reference distance, α is the path-loss exponent, $d_{CUE,BS}$ is the distance between the CUE and the BS, W is the communication bandwidth, N_0 is the one-sided spectral density of additive white Gaussian noise, and I_B is the cumulative interference at the BS from all DUE transmitters and we have

$$I_B = KP_D {d_0}^{\alpha} \sum_i {d_{DUE_{i\_TX},BS}}^{-\alpha} , \tag{5.23}$$

where $d_{DUE_{i\_TX},BS}$ is the distance between the DUE transmitter in the ith D2D pair and the BS.

Likewise, the received SINR at the DUE receiver of the ith D2D pair from the DUE transmitter in the same pair is

$$SINR_{Di} = \frac{KP_D {d_0}^{\alpha} {d_{DUE_{i\_TX},DUE_{i\_RX}}}^{-\alpha}}{N_0 W + I_{Di}} , \tag{5.24}$$

and the corresponding link capacity is

$$C_{Di} = W \log_2(1 + SINR_{Di}) , \tag{5.25}$$

where $d_{DUE_{i\_TX},DUE_{i\_RX}}$ is the distance between the DUE transmitter and receiver of the ith D2D pair, and I_{Di} is the cumulative interference received at the DUE receiver of the ith D2D pair, which consists of two parts, namely, the interference from the CUE, I'_{Di}, and the cumulative interference from all other D2D pairs except pair i, I''_{Di}, and we have

$$I_{Di} = I'_{Di} + I''_{Di} \ , \tag{5.26}$$

$$I'_{Di} = KP_C {d_0}^{\alpha} {d_{CUE,DUE_{i\_RX}}}^{-\alpha} \ , \tag{5.27}$$

$$I''_{Di} = KP_D {d_0}^{\alpha} \sum_{j \neq i} {d_{DUE_{j\_TX},DUE_{i\_RX}}}^{-\alpha} \ , \tag{5.28}$$

where $d_{CUE,DUE_{i\_RX}}$ is the distance between the CUE and the DUE receiver of the ith D2D pair, and $d_{DUE_{j\_TX},DUE_{i\_RX}}$ is the distance between the DUE receiver of the ith D2D pair and the DUE transmitter of a D2D pair other than i.

With the approach to Ran2Ran NDD proposed in Chapter 4 and the approach to Ref2Ran NDD proposed in [58], all the distributions of the distances involved in our model can be obtained, including

- the distribution of the distance from a reference point (e.g., the BS) to a random point (e.g., a CUE/DUE), for obtaining the distribution of the SINR at the BS,
- the distribution of the distance between two random points (e.g., DUE–DUE or CUE–DUE), for obtaining the distribution of the SINR at a DUE receiver, and
- the distribution of the SINR at either the BS or a DUE receiver based on the obtained distance distributions.

5.4.3 Approaches to Distance and SINR Distributions

According to our model, the RVs shown in Figure 5.7 for the distances in (5.21)–(5.28) are listed in Table 5.1, along with their cumulative distribution functions (CDFs). Depending on whether or not there is a GR set for the BS to guarantee its reception from the CUE, all the involved distances will have different distributions except $d_{CUE,BS}$, since the GR only affects the locations of DUEs. For the case without GR, considering the randomness of the locations of all the UEs in the cell, the distance between the CUE and BS and that between a DUE transmitter and the BS can be

Table 5.1: RVs and their CDFs for the distances in (5.21)–(5.28)

	Distance	**RV (CDF)**	
		No GR	**With GR**
SINR at the BS	$d_{CUE,BS}$ in (5.21)	X $(F_X(d))$	X $(F_X(d))$
	$d_{DUE_{i\_TX},BS}$ in (5.23)	X $(F_X(d))$	W $(F_W(d))$
SINR at a DUE receiver	$d_{DUE_{i\_TX},DUE_{i\_RX}}$ in (5.24)	Z $(F_Z(d))$	Q $(F_Q(d))$
	$d_{CUE,DUE_{i\_RX}}$ in (5.27)	Y $(F_Y(d))$	V $(F_V(d))$
	$d_{DUE_{j\_TX},DUE_{i\_RX}}$ in (5.28)	Y $(F_Y(d))$	U $(F_U(d))$

viewed as i.i.d. RVs (both denoted as X), and the same for the distance between the CUE and a DUE receiver and that between a DUE transmitter and a DUE receiver which are in different D2D pairs (both denoted as Y). For the two DUEs that can communicate with each other and form a D2D pair, the distance between them (RV Z or Q) has a different distribution from that of Y or U, due to the D2D communication range limit, i.e., $[d_{\min}^{D2D}, d_{\max}^{D2D}]$. As mentioned above, the SINR at the BS involves the distance from a reference point to a random point, and that at a DUE receiver involves the distance between two random points. With such a classification, we next show in detail how to obtain the CDFs of the above RVs.

Distance from a Reference Point to a Random Point

1) $F_X(d)$: As introduced in Chapter 4, $F_X(d)$ can be obtained by (4.1).

2) $F_W(d)$: Due to the effect of the GR for the BS, $F_W(d)$ is different from $F_X(d)$. Given that the GR has also a polygonal shape as shown in Figure 5.7b, we can first obtain the CDF of the distribution of the distance from the BS to a random point inside the GR, denoted as $F_G(d)$, by also using (4.1). Let S_{GR} denote the area of the GR and $S_{\widetilde{GR}}$ the area of the region (or ring, denoted as $\widetilde{\text{GR}}$) which is beyond the GR but within the cell. The area of the cell is S. The possibility that the random point appears in the GR is $\frac{S_{GR}}{S}$, while $\frac{S_{\widetilde{GR}}}{S}$ in the $\widetilde{\text{GR}}$. Therefore, with a weighted probabilistic sum, we have

$$F_X(d) = \frac{S_{GR}}{S} F_G(d) + \frac{S_{\widetilde{GR}}}{S} F_W(d) \ . \tag{5.29}$$

Thus, $F_W(d)$ can be obtained.

Note that if there are nodes deployed uniformly at random in the GR and $\widetilde{\text{GR}}$ but with different node densities, the above weighted probabilistic sum is still applicable, but with the weights modified correspondingly due to the node density differences. Specifically, assuming the node density ratio between the GR and $\widetilde{\text{GR}}$ is $\lambda_1 : \lambda_2$ (λ_1 and λ_2 are not zero at the same time),

$$F_X(d) = \frac{S_{GR}\lambda_1}{S_{GR}\lambda_1 + S_{\widetilde{GR}}\lambda_2} F_G(d) + \frac{S_{\widetilde{GR}}\lambda_2}{S_{GR}\lambda_1 + S_{\widetilde{GR}}\lambda_2} F_W(d) \ , \tag{5.30}$$

Especially, if $\lambda_1 : \lambda_2 = 1 : 1$, i.e., both areas have the same node density, then $F_X(d)$ is just what shown in (5.29). Therefore, the proposed approach in [58] has a potential of handling the networks with non-uniform nodal distributions.

Distance between Two Random Points

It becomes much more challenging to obtain the corresponding distance distribution due to the randomness of both points, especially for the case where the cell has an arbitrary irregular shape in our model, and a GR may be set for the BS, i.e., another arbitrarily-shaped void area inside (hereafter, we call a geometry containing a void area a "ring"). Nevertheless, with the approach proposed in Chapter 4, we can obtain the corresponding distance distributions.

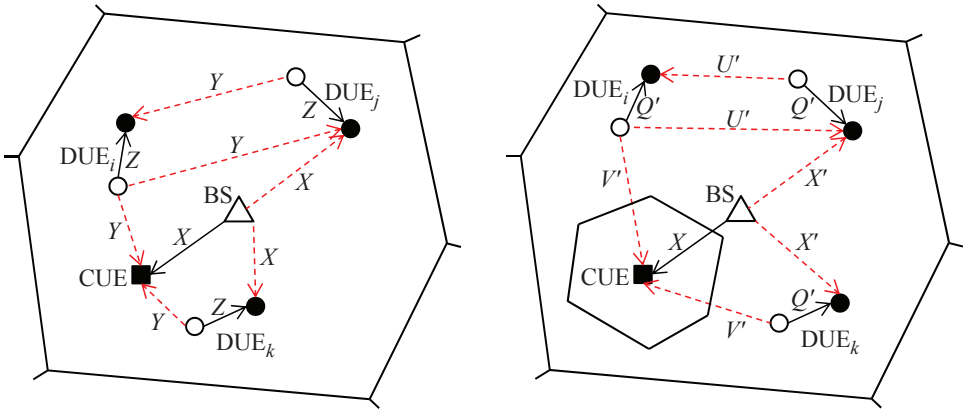
1) $F_Y(d)$ and $F_Z(d)$: $F_Y(d)$ in our model can be obtained according to (4.6). Meanwhile, $F_Z(d)$ can be obtained based on $F_Y(d)$ by considering the D2D communication range,

$$F_Z(d) = \frac{F_Y(d) - F_Y(d_{\min}^{D2D})}{F_Y(d_{\max}^{D2D}) - F_Y(d_{\min}^{D2D})} \ . \tag{5.31}$$

2) $F_U(d)$, $F_V(d)$ and $F_Q(d)$: $F_U(d)$ and $F_V(d)$ can be obtained through (4.10). By considering the D2D communication range, $F_Q(d)$ can be obtained based on $F_U(d)$ using (5.31). If there are nodes uniformly deployed in $\mathcal{K}_2$ and $\mathcal{K}_3$ but with different node densities, (4.11) should be applied.

Interference and SINR Distributions

The approach shown in Sections 5.3.2—"Interference Distribution" and 5.3.2—"SINR Distribution" can be utilized to obtain the corresponding interference and SINR distributions, respectively.



(a) No GR for the CUE (b) With GR for the CUE

Figure 5.8: System model for the downlink reusing mode. The RVs, X, Y, and Z , are the same as those shown in Figure 5.7.

5.4.4 Framework Applicability

Downlink Reusing Scenario

In an uplink reusing scenario, the D2D communications which are close to the BS may cause significant interference to the cellular communications due to the near-far effect [159]. Therefore, reusing downlink resources is also very important. The proposed framework can also be applied to a downlink reusing scenario, where concurrent D2D pairs will reuse the downlink resource of the BS, and thus the transmitting DUEs and BS can interfere with the reception at the CUE and the DUE receivers, respectively, as shown in Figure 5.8. Accordingly, the received SINR at the CUE from the BS is

$$SINR_C = \frac{K P_B {d_0}^{\alpha} {d_{CUE,BS}}^{-\alpha}}{N_0 W + I_C} \ , \tag{5.32}$$

where P_B is the transmission power of the BS, and I_C, the cumulative interference at the CUE, is

$$I_C = K P_D {d_0}^{\alpha} \sum_i {d_{DUE_{i\_TX},CUE}}^{-\alpha} \ . \tag{5.33}$$

Meanwhile, the received SINR at a DUE receiver in an arbitrary D2D pair, say pair i, can still be obtained according to (5.24), except that one of the two interference components in I_{Di}, i.e., I'_{Di} in (5.27) should be changed to

$$I'_{Di} = K P_B {d_0}^{\alpha} {d_{BS,DUE_{i\_RX}}}^{-\alpha} \ , \tag{5.34}$$

to represent the interference from the BS.

There are also two cases depending on whether a GR is set for the CUE to mitigate the interference from DUEs or not. For the case where there is no GR (e.g., Figure 5.8a), all the distributions of the distances involved can be obtained using the approaches explained in Section 5.4.3. As shown in Figure 5.8, the RVs, X, Y, and Z, are the same as those shown in Figure 5.7.

However, if there is a GR for the CUE and hence the D2D communications can only happen outside the GR (e.g., Figure 5.8b), it becomes much more challenging to obtain the distributions of the distances involved in (5.24), (5.34), and (5.28), i.e., $d_{DUE_{i\_TX},DUE_{i\_RX}}$, $d_{BS,DUE_{i\_RX}}$ and $d_{DUE_{j\_TX},DUE_{i\_RX}}$, respectively, due to the randomness of the location of the GR for the CUE. Thus, it is difficult to obtain the distribution of the received SINR at a DUE receiver. But the distribution of the distance, $d_{CUE,BS}$, is still the same as that in the no-guard-region scenario, since the GR only affects the distribution of DUEs. Meanwhile, in a simple assumption where the GR is approximated by a circle with radius of R and centered at the CUE, the CDF of $d_{DUE_{i\_TX},CUE}$, $F_{V'}(d)$, can be obtained based on $F_Y(d)$,

$$F_{V'}(d) = \frac{F_Y(d) - F_Y(R)}{1 - F_Y(R)} \ . \tag{5.35}$$

Therefore, the distribution of the received interference and SINR at the CUE can still be obtained.

Inter-cell Interference in A Multi-cell Network

If the frequency reuse factor in a cellular system is one, i.e., all cells are using the same resources, it needs to take into consideration the interference from the UEs in neighbor cells. Then, we need to obtain the distributions of the distances from a BS in a cell to a UE in a different cell and from a DUE in a cell to a UE in a different cell. To obtain the former, the approach proposed in [58] still applies. For the latter, our approach proposed in Chapter 4 also applies, as illustrated in Figure 4.3a. As a result, the proposed framework in this chapter can handle inter-cell interference in a multi-cell network.

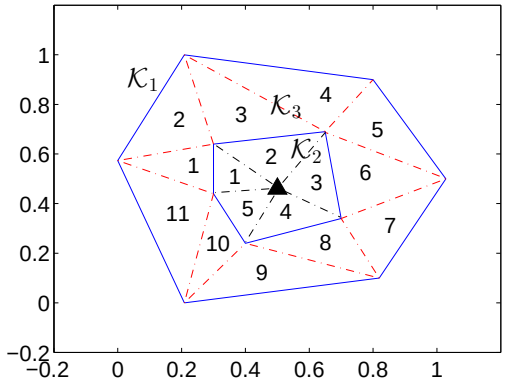


Figure 5.9: An example of irregular cell $\mathcal{K}_1$ and GR $\mathcal{K}_2$, with the ring area labeled by $\mathcal{K}_3$ (▲: BS).

Sector-Partitioned Cells

To increase the network capacity of a cellular system, equipping BSs with multiple directional antennas is a promising solution and has been actively studied and modeled for the throughput bound analysis on the underlaying D2D communications in [67]. With directional antennas, a cell can be divided into several smaller sections or so-called sectors, where it is possible for multiple CUEs to communicate with a BS simultaneously. Similarly, with the approaches proposed in [58] and Chapter 4 to the required distance distributions, the proposed framework can deal with sector-partitioned cells, whereby not only a single sector but also the inter-sector influence can be investigated, for both cellular and D2D communications, which would be a much further step compared to the D2D bound analysis in [67].

Irregular Cells and GRs

For the ease of presentation, in Section 5.4.5, the network performance is evaluated with a hexagonal cell and circular GR. However, as explained above, the proposed framework applies to the scenarios with irregular polygon-shaped cells and GRs. This is because both the underlying approaches to obtaining the distributions of the distance between a given Reference point and a Random one (Ref2Ran), and that between two Random points (Ran2Ran), can handle irregular polygons.

Figure 5.9 shows an example of irregular cell and GR for illustration. To obtain the required Ref2Rand distance distributions, as shown in Section 5.4.3, the triangulation with respect to the BS is adopted. For example, in Figure 5.9, the area

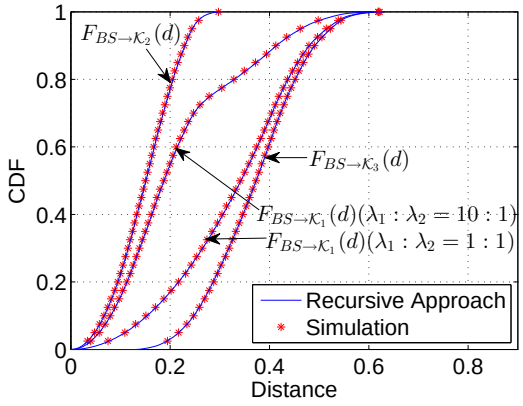


Figure 5.10: Distributions of the distance from the BS to a random point associated with the irregular cell and GR shown in Figure 5.9. $\lambda_1 : \lambda_2$ in (5.30) is set to $1:1$ or $10:1$.

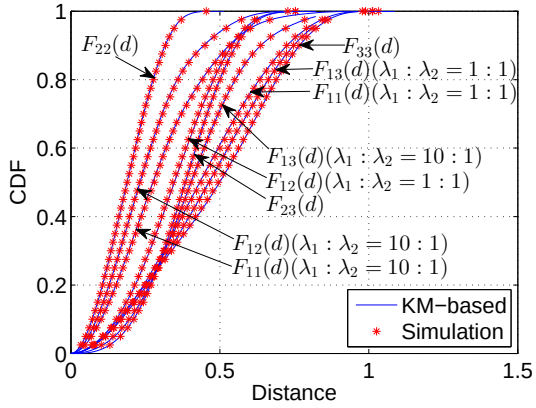


Figure 5.11: Distributions of the distance between two random points associated with the irregular cell and GR shown in Figure 5.9. $\lambda_1 : \lambda_2$ in (4.11) is set to $1:1$ or $10:1$.

$\mathcal{K}_2$ is triangulated with respect to the BS with dashed black lines into five triangles. Then based on (4.1), the distribution of the distance from the BS to a random point in $\mathcal{K}_2$ is obtained (denoted as $F_{BS\to\mathcal{K}_2}(d)$, i.e., $F_G(d)$). In the same way, $F_{BS\to\mathcal{K}_1}(d)$ (i.e., $F_X(d)$) can be obtained. Then based on (5.30), $F_{BS\to\mathcal{K}_3}(d)$ (i.e., $F_W(d)$) can be obtained, assuming that the CUE appears in the areas of GR and $\widetilde{\text{GR}}$ with the same possibility (i.e., $\lambda_1 : \lambda_2 = 1:1$ in (5.30)). The obtained results are shown in Figure 5.10.

To obtain the required Ran2Ran distance distributions, such as $F_{13}(d)$ (for obtaining $F_V(d)$) and $F_{33}(d)$ (for obtaining $F_U(d)$ and $F_Q(d)$), the ring area $\mathcal{K}_3$ is triangulated with dashed red lines into 11 triangles named by a number inside as shown

in Figure 5.9. As demonstrated in Chapter 4, with the proposed approach we can obtain Ran2Ran distance distributions associated with arbitrary triangles, including the Ran2Ran distance distribution within an arbitrary triangle and that between two arbitrary triangles. Then with a probabilistic sum,

$$F_{33}(d) = \sum_{i=1}^{11} \sum_{j=1}^{11} \frac{S_i S_j}{(\sum_{k=1}^{11} S_k)^2} F_{ij}(d) \ , \tag{5.36}$$

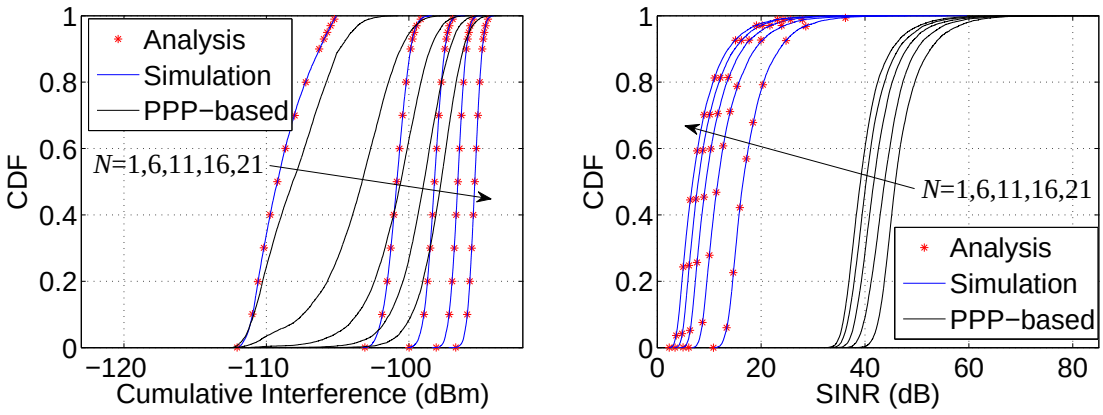
where S_x is the area of triangle x, and $F_{xy}(d)$ or $F_{yx}(d)$ is the CDF of the Ran2Ran distance distribution within triangle x if $x = y$, or between two triangles x and y if $x \neq y$. $F_{11}(d)$ and $F_{22}(d)$ can be obtained in the same way. Then based on (4.11), $F_{23}(d)$ and further $F_{13}(d)$ can be obtained, assuming that the CUE appears in the areas of GR and $\widetilde{\text{GR}}$ with the same possibility (i.e., $\lambda_1 : \lambda_2 = 1 : 1$ in (4.11)). The obtained results are shown in Figure 5.11.

Non-Uniform UE Distribution

If UEs are non-uniformly distributed in a cell, the cell area can be viewed in a different way, so that the non-uniform distribution can be considered as layers of uniform distributions, each of which can be addressed by the proposed framework. As an example, referring to the system model where a GR is set for the BS, DUEs are distributed in the cell but outside the GR. Therefore, the distribution of the DUEs is non-uniform in the entire cell, but is uniform in the cell minus the GR hole. On the other hand, it has a higher possibility that the UEs inside GR will choose the CUE mode than those inside $\widetilde{\text{GR}}$. Assuming such a probability ratio is $\lambda_1 : \lambda_2 = 10 : 1$, based on (5.30) and (4.11), we can obtain the corresponding distributions, which are different from those obtained when the probability ratio is $\lambda_1 : \lambda_2 = 1 : 1$, as shown in Figure 5.10 and Figure 5.11, respectively. Therefore, the proposed framework can also analyze the performance of the cellular network underlaying D2D communications with the non-uniform UE distribution described above.

5.4.5 Performance Evaluation

In this subsection, the proposed framework is evaluated in terms of two important network metrics: cumulative interference and outage probability. The obtained numerical results from the framework are verified in comparison with simulations.



(a) Interference distribution (b) SINR distribution

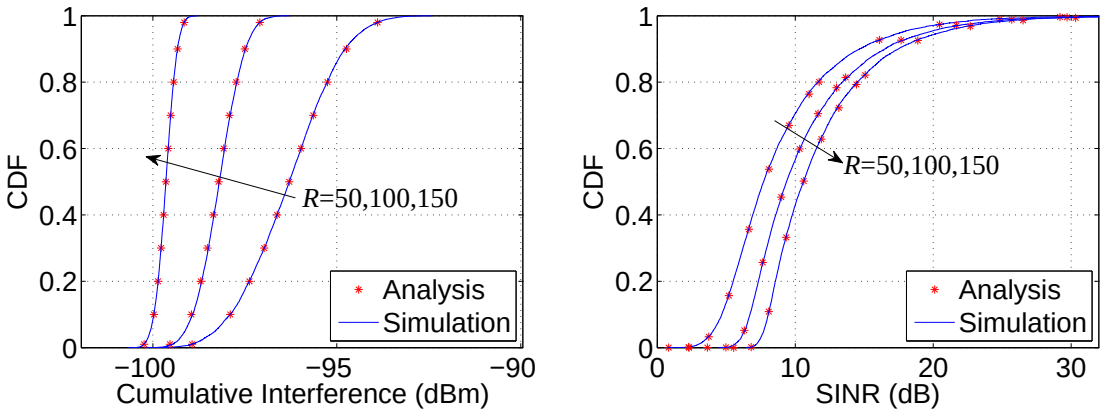
Figure 5.12: Distributions of the interference and SINR at the BS, with $R = 100$ m and $N = 1, 6, 11, 16,$ and 21.

Meanwhile, the effect of several system parameters on the performance metrics are investigated. The framework can provide insights when tuning the system parameters, e.g., the GR area for the BS and the number of concurrent D2D pairs, given the outage probability thresholds of both the BS and DUE receiver.

Considering the system model described in Section 5.4.2, for the ease of presentation and WLOG, the cell is approximated as a regular hexagon with side length of 225 m and the GR for the BS as a circle with radius of R and centered at the BS. Again, the proposed approach does not impose any constraints on the shapes of the cell or GR. $P_D = 2$ mWatt, $P_C = 0.2$ Watt, $d_{\min}^{D2D} = 2$ m, and $d_{\max}^{D2D} = 15$ m. The path-loss ratios are set as $L_{\mathrm{B}}(d) = -128.1 - 20\lg(d/1000)$ (dB) for the BS–UE link, and $L_{\mathrm{D}}(d) = -38 - 20\lg(d)$ (dB) for the DUE–DUE link [66]. Next, both the distributions of the cumulative interference and SINR at either the BS or a DUE receiver are shown, with the numerical results of our approach and the Matlab simulation results compared.

Distributions of Interference and SINR at the BS

We first evaluate the effect of the number of concurrent D2D pairs on the distributions of the interference and SINR received at the BS. With the GR radius R fixed to 100 m and the number of concurrent D2D pairs N varying from one to 21, the CDFs of the interference and SINR received at the BS are shown in Figure 5.12a and Figure 5.12b, respectively. Obviously, given that the size of the GR is the same,

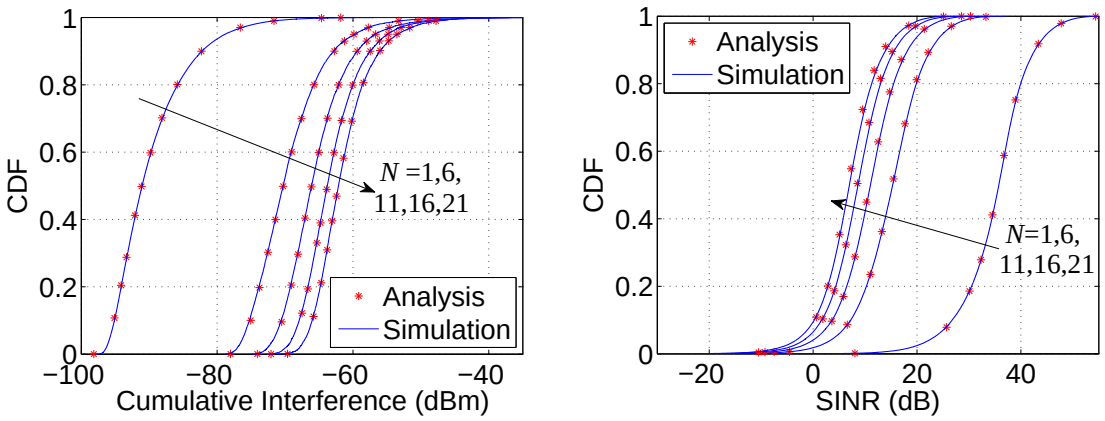


(a) Interference distribution (b) SINR distribution

Figure 5.13: Distributions of the interference and SINR at the BS, with $N = 11$ and $R = 50, 100,$ and 150 m.

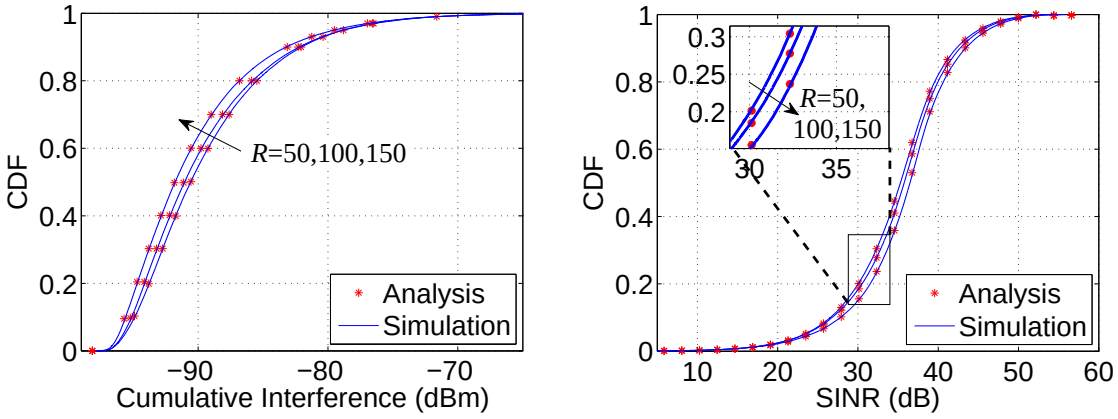
more D2D pairs transmitting simultaneously can impose a higher interference on the BS, and thus the received SINR at the BS becomes lower, which increases its outage probability. As a result, for the interference distribution shown in Figure 5.12a, the CDF curves located on the right-hand side correspond to a larger N; while for the SINR distribution shown in Figure 5.12b, the CDF curves located on the left-hand side correspond to a larger N. Moreover, a good match between the simulation and analytical results can be observed in both figures. The results obtained based on the PPP model with the same node density are also shown for comparison. The implementation of the PPP model in Matlab is similar to the simulation implementation shown in Section 5.3.3, except that the number of nodes over all simulations follows a Poisson distribution. The comparison shows large gaps for both the interference and SINR achieved at the BS. Due to the border effect, PPP is inapplicable for the system model of finite networks used in this section. In what follows, the comparison with PPP will be omitted.

The effect of the GR size is also investigated. Specifically, with the number of concurrent D2D pairs fixed to $N = 11$ and the radius of the GR R varying from 50 to 150 m, Figure 5.13a and Figure 5.13b show the corresponding interference and SINR distributions, respectively. Intuitively, as the area of the GR for the BS increases, DUE transmitters are "pushed" farther away from the BS, and their cumulative interference to the reception at the BS decreases, and thus the received SINR at the BS becomes high, which decreases the outage probability of the BS, as shown in the



(a) Interference distribution (b) SINR distribution

Figure 5.14: Distributions of the interference and SINR at a DUE receiver, with $R = 100$ m and $N = 1, 6, 11, 16,$ and 21.



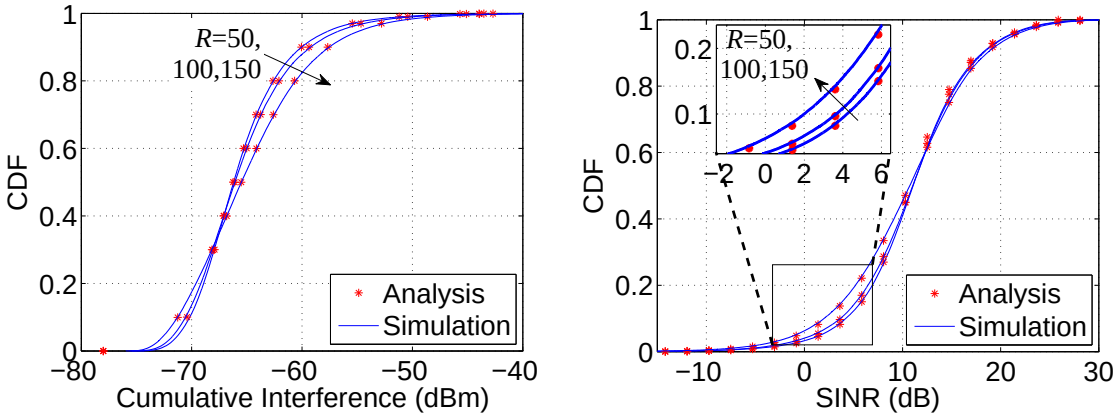
(a) Interference distribution ($N = 1$) (b) SINR distribution ($N = 1$)

Figure 5.15: Distributions of the interference and SINR at a DUE receiver, with $N = 1$ and $R = 50, 100,$ and 150 m.

figures.

Distributions of Interference and SINR at a DUE Receiver

In this subsection, the distributions of the interference and SINR received at a DUE receiver in an arbitrary D2D pair are investigated. First, with $R = 100$ m and N varying from one to 21, the corresponding CDFs are shown in Figure 5.14. When there is only one D2D pair, i.e., $N = 1$, the interference at the DUE receiver only comes from the CUE. As N increases, besides the interference from the CUE, a DUE



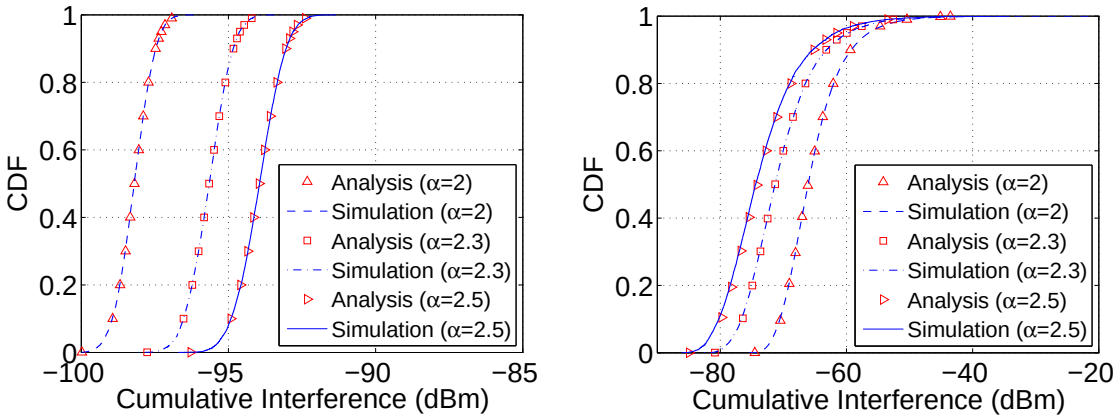
(a) Interference distribution ($N = 11$) (b) SINR distribution ($N = 11$)

Figure 5.16: Distributions of the interference and SINR at a DUE receiver, with $N = 11$ and $R = 50, 100,$ and 150 m.

receiver in a pair will also receive the interference from the DUE transmitters in all other D2D pairs, which leads to a shift of the CDF curves shown in Figure 5.14a towards the arrow direction. Thus, the received SINR decreases, which increases the outage probability of the D2D communications, as shown in Figure 5.14b.

The effect of the GR size on the D2D communications is then investigated. We first fix $N = 1$, so only the CUE will interfere with the reception at the DUE receiver. As R increases from 50 to 150 m, the probability that there is a large distance between the CUE and the DUE receiver becomes higher, and thus the interference received at the DUE receiver decreases, as shown in Figure 5.15a, which correspondingly increases its received SINR and thus decreases the outage probability of the D2D communication, as shown in Figure 5.15b. In addition, from Figure 5.15, we can observe that the GR size has a low impact on the D2D communication, as the DUE is less likely affected by a single CUE. Especially, when $R = 50$ and 100 m, the corresponding CDF curves of the SINR shown in Figure 5.15b almost overlap with each other.

Then, we fix $N = 11$, so besides the interference from the CUE, an arbitrary D2D pair will also receive the interference from other 10 D2D pairs. The results are shown in Figure 5.16. In contrast to the previous case where $N = 1$, as R increases, the cumulative interference to a D2D pair increases as well, as shown in Figure 5.16a, and thus the received SINR decreases, which increases the outage probability of D2D communications, as shown in Figure 5.16b. The results indicate that, with more concurrent D2D pairs, the interference to a D2D pair from all other D2D pairs will



(a) Distribution of the interference at the BS (b) Distribution of the interference at a DUE receiver

Figure 5.17: Interference distributions with $R = 100$ m, $N = 11$, and $\alpha = 2, 2.3,$ and 2.5.

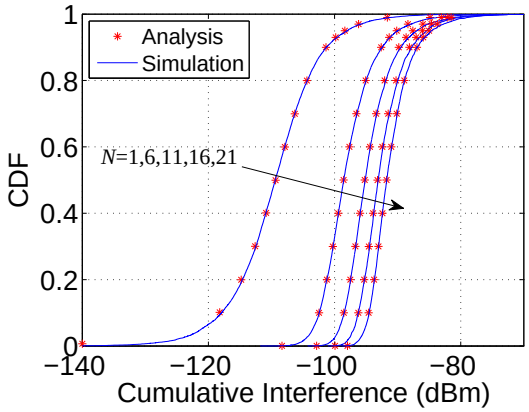


Figure 5.18: Distributions of the interference at the BS from DUE transmitters with the Rayleigh fading effect considered ($R = 0$ m and $N = 1, 6, 11, 16,$ and 21).

increase as the GR area increases, and such an increase dominates over the decrease of the interference from the CUE. Also, the CDFs for $R = 50$ and 100 m shown in Figure 5.16 almost overlap with each other when DUEs are still sparsely spread out in the cell.

Figure 5.17 shows that distributions of the interference at the BS (Figure 5.17a) and a DUE receiver (Figure 5.17b), with $R = 100$ m, $N = 11$, and path-loss exponent $\alpha = 2, 2.3,$ and 2.5. Since the BS–UE link and DUE–DUE link have different path-loss ratios due to antenna heights, the interference changes differently for the BS and a DUE receiver as α increases.

All the results shown above in a close match with the simulation results demonstrate the accuracy of our approach. Based on our performance studies, the proposed analytical framework in this section can be used to guide the planning and dimensioning at the system level, e.g., on average, how large N is allowed to be and how R is set by considering the bounded outage probabilities for both cellular and D2D communications systematically and comprehensively.

As mentioned in Section 5.2.2, the proposed model can be readily extended to include the shadowing and fading effects of wireless channels. Take the interference from DUE transmitters to the BS in the uplink resource reusing scenario for example. With the Rayleigh fading effect considered by using (5.5), Figure 5.18 shows the CDFs of the interference received at the BS.

5.5 Conclusions

This chapter applied NDD to carry out physical interference modeling and analysis in WANETs. We first analyzed an LSN running PDC. Different from the protocol modeling and analysis conducted in Chapter 3, the NDD-based modeling and analysis considers the accumulative interference at a receiver. Specifically, utilizing the distance distributions obtained in Chapter 4, we derived the distributions of interference and SINR. Given the outage probability of sensor nodes, the protocol parameters of PDC can be well tuned based on the obtained results. A similar framework is also applied for the performance analysis in a cellular network with underlaying ad hoc D2D communications. Based on the obtained results, the interference and outage probabilities for both the cellular and D2D communications in an uplink reusing scenario were thoroughly investigated. The accurate analysis demonstrated the promising potentials of the proposed framework, which is a significant complement to the existing approaches and results, and can provide meaningful insights and guidelines for the design and optimization of D2D communications in next-generation cellular networks.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this dissertation, we have investigated various important and fundamental aspects of protocol design and performance evaluation in WANETs. We first proposed a data collection protocol for WSNs. Then we modeled the proposed protocol in multi-hop LSNs. The model cannot consider the cumulative interference from all concurrent non-dedicated transmitters due to the assumption that each node has fixed interference and transmission ranges. Therefore, we further developed a physical interference model framework to consider cumulative interference. The framework relies on Ran2Ran NDD, which can be obtained with the systematic and unified approach proposed in this dissertation. Specifically, this dissertation contains:

1. **Design of a practical data collection protocol for duty-cycled ad hoc sensor networks**. We have proposed PDC which adopts a cross-layer integration design by only relying on PRTS/PCTS handshake. All the components of PDC are integrated in a natural and seamless way, including duty-cycled, pipelined scheduling, schedule synchronization, data-gathering tree establishment, topology control and maintenance, etc. We further propose ADC based on PDC with two additional features being integrated, i.e., free addressing and dynamic duty-cycling. Such integrations are beneficial to simplifying the protocol implementation and achieving high energy efficiency and low packet delivery latency, as well as improving network heterogeneity and load adaptivity. We have implemented both PDC and ADC in Contiki OS and built a testbed based on two hardware platforms to evaluate their performance.

2. **Modeling and analysis of a class of duty-cycled, multi-hop data collection protocols for LSNs**. The model consists of two parts, i.e., protocol process analysis and queueing behavior analysis. The model is general for both saturated and unsaturated scenarios, with and without retransmission. Different from existing models, the relaying nodes in our model can also generate their own packets as in reality. With the model, we have thoroughly investigated the PDC performance in an LSN in terms of the system throughput, energy efficiency, and packet delivery latency. The accuracy of the model has been validated under various scenarios compared with extensive OPNET simulations.

3. **A systematic and algorithmic approach to Ran2Ran NDD**. The approach is developed based on the concept, called kinematic measure in integral geometry. With decomposition and recursion methods, it can handle arbitrarily-shaped networks, including convex, concave, disjoint, and tiered networks, as well as non-uniform nodal distributions. Therefore, a wide range of new and significant NDD-based network modeling and performance analysis can be conducted. Based on the proposed approach, this dissertation has obtained and validated the NDDs associated with arbitrary polygons.

4. **Physical interference modeling and analysis based on NDD for LSNs and underlaying ad hoc D2D communications**. Different from the PDC design and modeling in Chapter 2 and Chapter 3, respectively, where location-based conditions are defined for successful communications between a pair of nodes, a physical interference model framework based on Ran2Ran NDD is proposed to analyze an LSN running the PDC protocol, taking into account the cumulative interference at a receiver. With the obtained results from the framework, the protocol parameters can be well tuned. In addition, utilizing both Ref2Ran and Ran2Ran NDDs, the framework is further utilized to analyze ad hoc D2D communications underlaying a cellular network. The proposed framework has no limitations on the network shapes. The presented results can provide important insights and guidelines for protocol parameter setting and network planning and dimensioning at the system level, by considering the bounded outage probabilities for inter-node communications systematically and comprehensively.

6.2 Future Work

For the future work that plans beyond this dissertation, there are still various open issues of importance. This section provides directions and ideas which we believe will appear and attract more attention.

1. In the current design of PDC, the sleeping schedules between two adjacent grades are staggered to achieve spatial diversity, so that inter-grade interference can be alleviated to some extent. The future work can also consider alleviating intra-grade interference by adopting time, spatial, and channel diversity to further improve the performance of the protocol for data collection. Although there have been several works recently utilizing multiple channels for data collection in WSNs [160–162], there is few work on comprehensively considering duty cycling, pipelined scheduling, schedule synchronization, and multi-channel communications, which we believe deserves more attention and critical study. Furthermore, the current PDC is designed based on omnidirectional antennas. It is also interesting to consider the design based on directional antennas, which can increase the potential of spatial reuse, provide longer transmission with the same amount of power and improve security [163].

2. In the work on modeling and analyzing a class of duty-cycled, multi-hop data collection protocols for LSNs in Chapter 3, we do not consider some practical issues in LSNs, such as the priority when queueing different packets (e.g., self-generated packets and forwarded packets), delay constraint/bound, energy balance, etc. In addition, the dynamic duty-cycling in ADC and multi-channel communication currently are not supported by the proposed model. Furthermore, we are also interested in comparing the modeling results with the experimental results. All these considerations can be included in the future work. Last but not least, a more general model is necessary for modeling a general multi-hop 2D network scenario. In general, either the model or the protocol could be improved in the future work.

3. NDD plays a significant role in the physical interference model framework proposed in Chapter 5. Although the relevant research on both types of NDD has achieved remarkable breakthroughs, there are still several open issues which are challenging to be solved. For both Ref2Ran and Ran2Ran NDDs, the approaches we proposed based on decomposition and recursion can handle the case where

nodes are uniformly distributed with different node densities in different subareas of a network, which leads to a discrete nonuniform node distribution. It is necessary to consider a more general, continuous nonuniform node distribution. In addition, the Ref2Ran NDDs can be easily extended to consider a reference point with height by the Pythagorean theorem; while for obtaining either Ref2Ran or Ran2Ran NDDs in a more general 3D scenario, there is few work on the approaches in the current literature.

Bibliography

[1] Lufeng Mo, Yuan He, Yunhao Liu, Jizhong Zhao, Shao-Jie Tang, Xiang-Yang Li, and Guojun Dai. Canopy closure estimates with GreenOrbs: Sustainable sensing in the forest. In *Proc. ACM SenSys*, pages 99–112, 2009.

[2] Shibo He, Jiming Chen, and Youxian Sun. Coverage and connectivity in duty-cycled wireless sensor networks for event monitoring. *Parallel and Distributed Systems, IEEE Transactions on*, 23(3):475–482, 2012.

[3] Fadi Al-Turjman, Hossam Hassanein, and Mohamed Ibnkahla. Efficient deployment of wireless sensor networks targeting environment monitoring applications. *Computer Communications*, 36(2):135–148, 2013.

[4] Ahmad El Kouche, Hossam Hassanein, and Khaled Obaia. Monitoring the reliability of industrial equipment using wireless sensor networks. In *Proc. IWCMC*, pages 88–93, 2012.

[5] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection tree protocol. In *Proc. ACM SenSys*, pages 1–14, 2009.

[6] Scott Moeller, Avinash Sridharan, Bhaskar Krishnamachari, and Omprakash Gnawali. Routing without routes: The backpressure collection protocol. In *Proc. ACM/IEEE IPSN*, pages 279–290, 2010.

[7] Yanwei Wu, Xiang-Yang Li, YunHao Liu, and Wei Lou. Energy-efficient wake-up scheduling for data collection and aggregation. *Parallel and Distributed Systems, IEEE Transactions on*, 21(2):275–287, 2010.

[8] Yongle Cao, Shuo Guo, and Tian He. Robust multi-pipeline scheduling in low-duty-cycle wireless sensor networks. In *Proc. IEEE INFOCOM*, pages 361–369, 2012.

[9] Imad Jawhar, Nader Mohamed, and Dharma P Agrawal. Linear wireless sensor networks: classification and applications. *Journal of Network and Computer Applications*, 34(5):1671–1682, 2011.

[10] Imad Jawhar, Mostafa Ammar, Sheng Zhang, Jie Wu, and Nader Mohamed. Ferry-based linear wireless sensor networks. In *Proc. IEEE GLOBECOM*, pages 304–309, 2013.

[11] Imad Jawhar, Nader Mohamed, Jameela Al-Jaroodi, and et al. A framework for using unmanned aerial vehicles for data collection in linear wireless sensor networks. *Journal of Intelligent & Robotic Systems*, 74(1-2):437–453, 2014.

[12] Hyung-Bong Lee, Ki-Hyeon Kwon, Lae-Jeong Park, and et al. A lightweight lap time measurement system for Alpine ski sport using a TDMA-based linear wireless sensor network. *Journal of Distributed Sensor Networks*, 2012.

[13] François Baccelli and Bartłomiej Błaszczyszyn. Stochastic geometry and wireless networks, volume I: Theory. 2009.

[14] Steven Weber, Jeffrey G Andrews, and Nihar Jindal. An overview of the transmission capacity of wireless networks. *Communications, IEEE Transactions on*, 58(12):3593–3604, 2010.

[15] Shizhen Zhao, Luoyi Fu, Xinbing Wang, and Qian Zhang. Fundamental relationship between node density and delay in wireless ad hoc networks with unreliable links. In *Proc. ACM MobiCom*, pages 337–348, 2011.

[16] Wei Ren, Qing Zhao, and Ananthram Swami. Temporal traffic dynamics improve the connectivity of ad hoc cognitive radio networks. *Networking, IEEE/ACM Transactions on*, 22(1):124–136, 2014.

[17] Sunil Srinivasa and Martin Haenggi. Distance distributions in finite uniformly random networks: theory and applications. *Vehicular Technology, IEEE Transactions on*, 59(2):940–949, 2010.

[18] Don Torrieri and Matthew C Valenti. The outage probability of a finite ad hoc network in Nakagami fading. *Communications, IEEE Transactions on*, 60(11):3509–3518, 2012.

[19] Matthew C Valenti, Don Torrieri, and Salvatore Talarico. A direct approach to computing spatially averaged outage probability. *IEEE Communications Letters*, 18(7):1103–1106, 2014.

[20] Maryam Ahmadi, Fei Tong, Lei Zheng, and Jianping Pan. Performance analysis for two-tier cellular systems based on probabilistic distance models. In *Proc. IEEE INFOCOM*, pages 352–360, 2015.

[21] Vahid Naghshin, Amir Masoud Rabiei, Norman C Beaulieu, and et al. Accurate statistical analysis of a single interference in random networks with uniformly distributed nodes. *IEEE Communications Letters*, 18(2):197–200, 2014.

[22] Pingyi Fan, Guansheng Li, Kai Cai, and et al. On the geometrical characteristic of wireless ad-hoc networks and its application in network performance analysis. *Wireless Communications, IEEE Transactions on*, 6(4):1256–1265, 2007.

[23] Zubin Bharucha and Harald Haas. The distribution of path losses for uniformly distributed nodes in a circle. *Research Letters in Communications*, 2008(4):1–4, 2008.

[24] Leonard E Miller. Distribution of link distances in a wireless network. *Journal of Research of the National Institute of Standards and Technology*, 106(2):401–412, 2001.

[25] Christian Bettstetter and Jorg Eberspacher. Hop distances in homogeneous ad hoc networks. In *Proc. IEEE VTC*, pages 2286–2290, 2003.

[26] Rodrigo S.C. Leão and Valmir C. Barbosa. Exploiting the distribution of distances between nodes to efficiently solve the localization problem in wireless sensor networks. In *Proc. ACM* PM^2HW^2N, pages 9–16, 2010.

[27] Yanyan Zhuang, Jianping Pan, and Lin Cai. Minimizing energy consumption with probabilistic distance models in wireless sensor networks. In *Proc. IEEE INFOCOM*, pages 1–9, 2010.

[28] Liang He, Zhe Yang, Jianping Pan, Lin Cai, and Jingdong Xu. Evaluating service disciplines for mobile elements in wireless ad hoc sensor networks. In *Proc. IEEE INFOCOM*, pages 576–584, 2012.

[29] Liang He, Zhe Yang, Jianping Pan, Lin Cai, Jingdong Xu, and Yu Gu. Evaluating service disciplines for on-demand mobile data collection in sensor networks. *Mobile Computing, IEEE Transactions on*, 13(4):797–810, 2014.

[30] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 36(SI):147–163, 2002.

[31] Mikls Marti, Branislav Kusy, Gyula Simon, and Ákos Ldeczi. The flooding time synchronization protocol. In *Proc. ACM SenSys*, pages 39–49, 2004.

[32] Jiming Chen, Qing Yu, Yan Zhang, and et al. Feedback-based clock synchronization in wireless sensor networks: A control theoretic approach. *Vehicular Technology, IEEE Transactions on*, 59(6):2963–2973, 2010.

[33] Jianping He, Peng Cheng, Ling Shi, Jiming Chen, and Youxian Sun. Time synchronization in WSNs: A maximum-value-based consensus approach. *Automatic Control, IEEE Transactions on*, 59(3):660–675, 2014.

[34] Jianping He, Jiming Chen, Peng Cheng, and et al. Secure time synchronization in wireless sensor networks: A maximum consensus-based approach. *Parallel and Distributed Systems, IEEE Transactions on*, 25(4):1055–1065, 2014.

[35] Hao Huang and Jihoon Yun. Scalable clock synchronization in wireless networks with low-duty-cycle radio operations. In *Proc. IEEE INFOCOM*, pages 2011–2019, 2015.

[36] Contiki Operating System. `http://www.contiki-os.org`.

[37] Zolertia. `http://zolertia.io/`.

[38] MicaZ. `http://www.capsil.org/capsilwiki/index.php/MICAz`.

[39] Change the default MAC address of Zolertia Z1. `http://zolertia.sourceforge.net/wiki/index.php/Mainpage:Contiki_apps#Change_the_default_MAC_address_and_Node_ID.2C_and_burn_it_to_flash`.

[40] Tmote Sky. `http://tmote-sky.blogspot.ca/`.

[41] EXP5438. `http://www.ti.com/tool/msp-exp430f5438`.

[42] Pekka Pirinen. Outage analysis of ultra-wideband system in lognormal multipath fading and square-shaped cellular configurations. *EURASIP Journal on Wireless Communications and Networking*, 2006:1–10, 2006.

[43] Konstantinos B. Baltzis. Analytical and closed-form expressions for the distribution of path loss in hexagonal cellular networks. *Wireless Personal Communications*, 60(4):599–610, 2011.

[44] Yanyan Zhuang and Jianping Pan. Random distances associated with rhombuses. *arXiv:1106.1257*, pages 1–14, 2011.

[45] Yanyan Zhuang and Jianping Pan. Random distances associated with hexagons. *arXiv:1106.2200*, pages 1–26, 2011.

[46] Konstantinos B Baltzis. A geometric method for computing the nodal distance distribution in mobile networks. *Progress In Electromagnetics Research*, 114:159–175, 2011.

[47] Y. Zhuang, Y. Luo, L. Cai, and J. Pan. A geometric probability model for capacity analysis and interference estimation in wireless mobile cellular systems. In *Proc. IEEE GLOBECOM*, pages 1–6, 2011.

[48] Yanyan Zhuang and Jianping Pan. A geometrical probability approach to location-critical network performance metrics. In *Proc. IEEE INFOCOM*, pages 1817–1825, 2012.

[49] Yanyan Zhuang and Jianping Pan. Random distances associated with equilateral triangles. *arXiv:1207.1511*, pages 1–13, 2012.

[50] Maryam Ahmadi and Jianping Pan. Random distances associated with trapezoids. *arXiv:1307.1444*, pages 1–12, 2013.

[51] Dmitri Moltchanov. Distance distributions in random networks. *Ad Hoc Networks*, 10(6):1146–1166, 2012.

[52] Uwe Bäsel. The distribution function of the distance between two random points in a right-angled triangle. *arXiv:1208.6228*, pages 1–15, 2012.

[53] Konstantinos B Baltzis. Distance distribution in convex n-gons: Mathematical framework and wireless networking applications. *Wireless Personal Communications*, 71(2):1487–1503, 2013.

[54] Zubair Khalid and Salman Durrani. Distance distributions in regular polygons. *Vehicular Technology, IEEE Transactions on*, 62(5):2363–2368, 2013.

[55] Fei Tong, Maryam Ahmadi, and Jianping Pan. Random distances associated with arbitrary triangles: a systematic approach between two random points. *arXiv:1312.2498*, pages 1–22, 2013.

[56] Uwe Bäsel. Random chords and point distances in regular polygons. *Acta Mathematica Universitatis Comenianae*, 83(1):1–18, 2014.

[57] Fei Tong, Maryam Ahmadi, Jianping Pan, Lei Zheng, and Lin Cai. Poster: Geometrical distance distribution for modeling performance metrics in wireless communication networks. In *Proc. ACM MobiCom*, pages 341–343, 2014.

[58] Maryam Ahmadi and Jianping Pan. Random distances associated with arbitrary triangles: a recursive approach with an arbitrary reference point. *UVic-Space*, pages 1–13, 2014.

[59] Ross Pure and Salman Durrani. Computing exact closed-form distance distributions in arbitrarily shaped polygons with arbitrary reference point. *Mathematica Journal*, 17:1–27, 2015.

[60] Sinan Sinanovic, Nikola Serafimovski, Harald Haas, and et al. Maximising the system spectral efficiency in a decentralised 2-link wireless network. *Wireless Communications Networking, EURASIP Journal on*, 2008:24:1–24:13, 2008.

[61] Luis A Santaló. *Integral Geometry and Geometric Probability*. Cambridge University Press, 2004.

[62] Arash Asadi, Qing Wang, and Vincenzo Mancuso. A survey on device-to-device communication in cellular networks. *IEEE Communications Surveys & Tutorials*, 16(4):1801–1819, 2014.

[63] Yongsheng Cheng, Han Han, and Xiaokang Lin. Device-to-device communication in CDMA-based cellular systems-uplink capacity analysis. In *Proc. IEEE ICCSN*, pages 430–434, 2011.

[64] Hyunkee Min, Jemin Lee, Sungsoo Park, and Daesik Hong. Capacity enhancement using an interference limited area for device-to-device uplink un-

derlaying cellular networks. *Wireless Communications, IEEE Transactions on*, 10(12):3995–4000, 2011.

[65] Rui Chen, Xuewen Liao, Shihua Zhu, and Zhonghua Liang. Capacity analysis of device-to-device resource reusing modes for cellular networks. In *Proc. IEEE ComNetSat*, pages 64–68, 2012.

[66] Minming Ni, Lei Zheng, Fei Tong, and et al. A geometrical-based throughput bound analysis for device-to-device communications in cellular networks. *Selected Areas in Communications, IEEE Journal on*, 33(1):100–110, 2015.

[67] Minming Ni, Jianping Pan, and Lin Cai. Geometrical-based throughput analysis of device-to-device communications in a sector-partitioned cell. *Wireless Communications, IEEE Transactions on*, 14(4):2232–2244, 2015.

[68] Chuan Ma, Weijie Wu, Ying Cui, and Xinbing Wang. On the performance of successive interference cancellation in D2D-enabled cellular networks. In *Proc. IEEE INFOCOM*, pages 37–45, 2015.

[69] Xiaohui Xu, Haibo Wang, Hao Feng, and Chengwen Xing. Analysis of device-to-device communications with exclusion regions underlaying 5G networks. *Emerging Telecommunications Technologies, Transactions on*, 26(1):93–101, 2015.

[70] Junchao Li, Weiwei Xia, and Lianfeng Shen. Delay-aware resource control for device-to-device underlay communication systems. *Emerging Telecommunications Technologies, Transactions on*, pages 1–14, 2015.

[71] Jiaheng Wang, Daohua Zhu, Chunming Zhao, J.C.F. Li, and Ming Lei. Resource sharing of underlaying device-to-device and uplink cellular communications. *IEEE Communications Letters*, 17(6):1148–1151, 2013.

[72] Daquan Feng, Lu Lu, Yi Yuan-Wu, G.Y. Li, Gang Feng, and Shaoqian Li. Device-to-device communications underlaying cellular networks. *Communications, IEEE Transactions on*, 61(8):3541–3551, 2013.

[73] Fei Tong, Minming Ni, Lei Shu, and Jianping Pan. A pipelined-forwarding, routing-integrated and effectively-identifying MAC for large-scale WSN. In *Proc. IEEE GLOBECOM*, pages 225–230, 2013.

[74] Fei Tong, Ruonan Zhang, and Jianping Pan. One handshake can achieve more: An energy-efficient, practical pipelined data collection for duty-cycled sensor networks. *IEEE Sensors Journal*, 16(9):3308–3322, 2016.

[75] Fei Tong and Jianping Pan. Adaptive data collection with free addressing and dynamic duty-cycling for sensor networks. In *Proc. EAI QShine*, 2016.

[76] Fei Tong and Jianping Pan. ADC: An adaptive data collection protocol with free addressing and dynamic duty-cycling for sensor networks. *Submitted to ACM/Springer Mobile Networks & Applications*, 2016.

[77] Fei Tong, Lei Zheng, Maryam Ahmadi, Minming Ni, and Jianping Pan. Modeling duty-cycling MAC protocols with pipelined scheduling for linear sensor networks. In *Proc. IEEE ICCC*, pages 813–817, 2014.

[78] Fei Tong, Lei Zheng, Maryam Ahmadi, Mingming Ni, and Jianping Pan. Modeling and analyzing duty-cycling pipelined-scheduling MAC for linear sensor networks. *Vehicular Technology, IEEE Transactions on*, 65(4):2608–2620, 2016.

[79] Fei Tong and Jianping Pan. Random distances associated with arbitrary polygons: a systematic approach between two random points. *arXiv:1602.03407*, pages 1–16, 2016.

[80] Fei Tong, Ruonan Zhang, and Jianping Pan. Distance distributions in finite ad hoc networks: Approaches, applications, and directions. In *Proc. EAI Adhocnets*, 2016.

[81] Fei Tong, Yin Wan, Lei Zheng, Jianping Pan, and Lin Cai. A probabilistic distance-based modeling and analysis for cellular networks with underlaying device-to-device communications. *Accepted by Wireless Communications, IEEE Transactions on*, 2016.

[82] Qing Cao, T. Abdelzaher, Tian He, and J. Stankovic. Towards optimal sleep scheduling in sensor networks for rare-event detection. In *Proc. ACM/IEEE IPSN*, pages 20–27, 2005.

[83] Shu Du, Amit Kumar Saha, and David B Johnson. RMAC: A routing-enhanced duty-cycle MAC protocol for wireless sensor networks. In *Proc. IEEE INFOCOM*, pages 1478–1486, 2007.

[84] Gang Lu, Bhaskar Krishnamachari, and Cauligi S Raghavendra. An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks. In *Proc. IEEE IPDPS*, pages 224–231, 2004.

[85] Antonio G. Ruzzelli, Gregory M.P. OHare, and Raja Jurdak. MERLIN: Cross-layer integration of MAC and routing for low duty-cycle sensor networks. *Ad Hoc Networks*, 6(8):1238–1257, 2008.

[86] Jia Li, Song Min Kim, and Tian He. Circular pipelining: Minimizing round-trip delay in low-duty-cycle wireless networks. In *Proc. IEEE ICNP*, pages 421–432, 2014.

[87] Hao Huang, Jihoon Yun, Ziguo Zhong, Songmin Kim, and Tian He. PSR: Practical synchronous rendezvous in low-duty-cycle wireless networks. In *Proc. IEEE INFOCOM*, pages 2661–2669, 2013.

[88] Tao Zheng, Sridhar Radhakrishnan, and Venkatesh Sarangan. PMAC: an adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proc. IEEE IPDPS*, pages 65–72, 2005.

[89] Xiaodong Wang, Xiaorui Wang, Liu Liu, and Guoliang Xing. DutyCon: A dynamic duty-cycle control approach to end-to-end delay guarantees in wireless sensor networks. *Sensor Networks, ACM Transactions on*, 9(4):1–33, 2013.

[90] Jianhui Zhang, Zhi Li, Feng Xia, Shaojie Tang, Xingfa Shen, and Bei Zhao. Cooperative scheduling for adaptive duty cycling in asynchronous sensor networks. *The Computer Journal*, pages 1–13, 2014.

[91] Affoua Thérese Aby, Alexandre Guitton, Pascal Lafourcade, and Michel Misson. SLACK-MAC: Adaptive MAC protocol for low duty-cycle wireless sensor networks. In *Ad Hoc Networks*, pages 69–81. 2015.

[92] Jeremy Elson and Deborah Estrin. Random, ephemeral transaction identifiers in dynamic sensor networks. In *Proc. ICDCS*, pages 459–468, 2001.

[93] James Jobin, Zhenqiang Ye, Honomount Rawat, Srikanth V Krishnamurthy, and Satish K Tripathi. A lightweight framework for source-to-sink data transfer in wireless sensor networks. In *Proc. BroadNets*, pages 703–713, 2005.

[94] Dazhi Chen, Jing Deng, and Pramod K Varshney. Efficient data delivery over address-free wireless sensor networks. In *Proc. CISS*, pages 16–18, 2005.

[95] Sunil Kulkarni, Aravind Iyer, and Catherine Rosenberg. An address-light, integrated MAC and routing protocol for wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 14(4):793–806, 2006.

[96] Weiwei Fang, Yi Liu, and Depei Qian. EDDS: An efficient data delivery scheme for address-free wireless sensor networks. In *Proc. ICN*, pages 1–7, 2007.

[97] Zhe Yang, Lin Cai, Yu Liu, and Jianping Pan. Environment-aware clock skew estimation and synchronization for wireless sensor networks. In *Proc. IEEE INFOCOM*, pages 1017–1025, 2012.

[98] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 12(3):493–506, 2004.

[99] Tijs Van Dam and Koen Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proc. ACM SenSys*, pages 171–180, 2003.

[100] Kaixin Xu, M. Gerla, and Sang Bae. How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks. In *Proc. IEEE GLOBECOM*, pages 72–76, 2002.

[101] Nicolas Burri, Pascal Von Rickenbach, and Roger Wattenhofer. Dozer: Ultra-low power data gathering in sensor networks. In *Proc. ACM/IEEE IPSN*, pages 450–459, 2007.

[102] Geoff Werner-Allen, Konrad Lorincz, Jeff Johnson, Jonathan Lees, and Matt Welsh. Fidelity and yield in a volcano monitoring sensor network. In *Proc. USENIX OSDI*, pages 381–396, 2006.

[103] Contiki Supported Platforms. `http://www.contiki-os.org/hardware.html`.

[104] Fredrik Osterlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, and Thiemo Voigt. Cross-level sensor network simulation with Cooja. In *Proc. IEEE Conference on Local Computer Networks*, pages 641–648, 2006.

[105] Joakim Eriksson, Fredrik Österlind, Niclas Finne, and et al. COOJA/MSPSim: Interoperability testing for wireless sensor networks. In *Proc. Simutools*, pages 27:1–27:7, 2009.

[106] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proc. ACM SenSys*, pages 307–320, 2006.

[107] PDC and ADC implementation in the Contiki Operating System. `https://github.com/fei-tong/PDC-ADC-in-Contiki`.

[108] Imad Jawhar, Nader Mohamed, Jameela Al-Jaroodi, and Sheng Zhang. An efficient framework for autonomous underwater vehicle extended sensor networks for pipeline monitoring. In *Proc. IEEE ROSE*, pages 124–129, 2013.

[109] Marco Zimmerling, Waltenegus Dargie, and Johnathan M Reason. Energy-efficient routing in linear wireless sensor networks. In *Proc. IEEE MASS*, pages 1–3, 2007.

[110] Marco Zimmerling, Waltenegus Dargie, and Johnathan M. Reason. Localized power-aware routing in linear wireless sensor networks. In *Proc. ACM International Conference on Context-awareness for Self-managing Systems*, pages 24–33, 2008.

[111] Imad Jawhar, Nader Mohamed, and Liren Zhang. A distributed topology discovery algorithm for linear sensor networks. In *Proc. IEEE ICCC*, pages 775–780, 2012.

[112] Brian Phelan, Peter Terlecky, Amotz Bar-Noy, Theodore Brown, and Dror Rawitz. Should I stay or should I go? maximizing lifetime with relays. In *Proc. IEEE DCOSS*, pages 1–8, 2012.

[113] Nader Mohamed, Jameela Al-Jaroodi, Imad Jawhar, and Abdulla Eid. Using mobile sensors to enhance coverage in linear wireless sensor networks. In *Proc. IEEE NCA*, pages 1–6, 2013.

[114] Nader Mohamed, Jameela Al-Jaroodi, Imad Jawhar, and Sanja Lazarova-Molnar. Failure impact on coverage in linear wireless sensor networks. In *Proc. SPECTS*, pages 188–195, 2013.

[115] Nader Mohamed, Haya AlDhaheri, Khadeijah Almurshidi, Mahra Al-Hammoudi, Salama Al-Yalyali, Imad Jawhar, and Jameela Al-Jaroodi. Using UAVs to secure linear wireless sensor networks. In *Proc. IEEE BigDataSecurity, IEEE HPSC, and IEEE IDS*, pages 424–429, 2016.

[116] An-Feng Liu, Xian-You Wu, Zhi-Gang Chen, and Wei-Hua Gui. An energy-balanced data gathering algorithm for linear wireless sensor networks. *International Journal of Wireless Information Networks*, 17(1-2):42–53, 2010.

[117] Daniele De Caneva and Pier Luca Montessoro. A synchronous and deterministic MAC protocol for wireless communications on linear topologies. *International Journal of Communications, Network & System Sciences*, 3(12):925–933, 2010.

[118] Chiara Buratti and Roberto Verdone. P-CSMA: A priority-based CSMA protocol for multi-hop linear wireless networks. In *Proc. European Wireless Conference*, pages 1–8, 2013.

[119] Zhong Shen, Hai Jiang, and Zhongjiang Yan. Fast data collection in linear duty-cycled wireless sensor networks. *Vehicular Technology, IEEE Transactions on*, 63(4):1951–1957, 2014.

[120] Petar MomČilović and Mark S. Squillante. On throughput in linear wireless networks. In *Proc. ACM MobiHoc*, pages 199–208, 2008.

[121] Moslem Noori and Masoud Ardakani. Characterizing the traffic distribution in linear wireless sensor networks. *IEEE Communications Letters*, 12(8):554–556, 2008.

[122] Nader Mohamed, Jameela Al-Jaroodi, Imad Jawhar, and Abdulla Eid. Reliability analysis of linear wireless sensor networks. In *Proc. IEEE NCA*, pages 11–16, 2013.

[123] Nader Mohamed, Jameela Al-Jaroodi, and Imad Jawhar. Modeling the performance of faulty linear wireless sensor networks. *International Journal of Distributed Sensor Networks*, pages 1–12, 2014.

[124] Ye Zhang, He Chen, and Jiang Lingge. Energy and QoS trade-off analysis of S-MAC protocol in wireless sensor networks. In *Proc. IET CCWMSN*, pages 76–79, 2007.

[125] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 12(3):493–506, 2004.

[126] Ou Yang and Wendi B Heinzelman. Modeling and throughput analysis for SMAC with a finite queue capacity. In *Proc. ISSNIP*, pages 409–414, 2009.

[127] Ou Yang and Wendi B Heinzelman. Modeling and throughput analysis for X-MAC with a finite queue capacity. In *Proc. IEEE GLOBECOM*, pages 1–5, 2010.

[128] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proc. ACM SenSys*, pages 307–320, 2006.

[129] Ou Yang and Wendi B Heinzelman. Modeling and performance analysis for duty-cycled MAC protocols with applications to S-MAC and X-MAC. *Mobile Computing, IEEE Transactions on*, 11(6):905–921, 2012.

[130] Mehmet Yunus Donmez, Sinan Isik, and Cem Ersoy. Combined analysis of contention window size and duty cycle for throughput and energy optimization in wireless sensor networks. *Computer Networks*, 57(5):1101–1112, 2013.

[131] Lakshmikanth Guntupalli, Jorge Martinez-Bauset, Frank Yong Li, and Mary Ann Weitnauer. Aggregated packet transmission in duty-cycled WSNs: Modeling and performance evaluation. *Vehicular Technology, IEEE Transactions on*, PP(99):1–1, 2016.

[132] Hyungkeun Lee and Hyukjoon Lee. Modeling and anaysis of an energy-efficient MAC protocol for wireless sensor networks. In *Proc. ICOIN*, pages 402–405, 2016.

[133] Inhye Park, Hyungkeun Lee, and Seokjoong Kang. RIX-MAC: An energy-efficient receiver-initiated wakeup MAC protocol for WSNs. *Internet and Information System, KSII Transactions on*, 8(5):1604–1617, 2014.

[134] Jun Luo, Lingge Jiang, and Chen He. Performance analysis of synchronous wakeup patterns in contention-based sensor networks using a finite queuing model. In *Proc. IEEE GLOBECOM*, pages 1334–1338, 2007.

[135] Jun Luo, Lingge Jiang, and Chen He. Finite queuing model analysis for energy and QoS tradeoff in contention-based wireless sensor networks. In *Proc. IEEE ICC*, pages 3901–3906, 2007.

[136] Jun Luo, LingGe Jiang, and Chen He. An analytical model for SMAC protocol in multi-hop wireless sensor networks. *Science China Information Sciences*, 53(11):2323–2331, 2010.

[137] Gang Lu, Bhaskar Krishnamachari, and Cauligi S. Raghavendra. An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks. *Wireless Communications & Mobile Computing*, 7(7):863–875, 2007.

[138] Tao Zheng, Sridhar Radhakrishnan, and Venkatesh Sarangan. Modeling and performance analysis of DMAC for wireless sensor networks. In *Proc. ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 119–128, 2011.

[139] Kishor S Trivedi. *Probability & statistics with reliability, queuing and computer science applications*. John Wiley & Sons, 2008.

[140] The implementation of PDC in OPNET. `https://github.com/fei-tong/PDC-in-OPNET`.

[141] P. Patil and U. Kulkarni. SVM based data redundancy elimination for data aggregation in wireless sensor networks. In *Proc. of Advances in Computing, Communications and Informatics (ICACCI), IEEE*, pages 1309–1316, 2013.

[142] Paulo Cardieri. Modeling interference in wireless ad hoc networks. *IEEE Communications Surveys & Tutorials*, 12(4):551–572, 2010.

[143] Martin Haenggi. On distances in uniformly random networks. *Information Theory, IEEE Transactions on*, 51(10):3584–3586, 2005.

[144] Sunil Srinivasa and Martin Haenggi. Modeling interference in finite uniformly random networks. In *International Workshop on Information Theory for Sensor Networks (WITS07)*, pages 1–12, 2007.

[145] Jing Guo, Salman Durrani, and Xiangyun Zhou. Outage probability in arbitrarily-shaped finite wireless networks. *Communications, IEEE Transactions on*, 62(2):699–712, 2014.

[146] Maryam Ahmadi, Minming Ni, and Jianping Pan. A geometrical probability-based approach towards the analysis of uplink inter-cell interference. In *Proc. IEEE GLOBECOM*, pages 4952–4957, 2013.

[147] Lingxiong Hu. The research to chord length of convex domain and the number of solutions to Neumann boundary value problem. Master's thesis, Wuhan University of Science and Technology, 2012.

[148] Delin Ren. Random chord distributions and containment functions. *Advances in Applied Math*, 58:1–20, 2014.

[149] Frank Piefke. Beziehungen zwischen der sehnenlängenverteilung und der verteilung des abstandes zweier zufälliger punkte im eikörper. *Probability Theory and Related Fields*, 43(2):129–134, 1978.

[150] Konstantinos B Baltzis. Spatial characterization of the uplink inter-cell interference in polygonal-shaped wireless networks. *Radioengineering*, 22(1):363–370, 2013.

[151] Olli Apilo, Mika Lasanen, Sandrine Boumard, and Aarne Mammela. The distribution of link distances in distributed multiple-input multiple-output cellular systems. In *Proc. IEEE VTC*, pages 1–5, 2013.

[152] Liang Cheng Shiu, Chao Yang Lee, and Chu Sing Yang. The divide-and-conquer deployment algorithm based on triangles for wireless sensor networks. *IEEE Sensors Journal*, 11(3):781–790, 2011.

[153] Zhen Wang, Xingzai Lv, and Jinkang Zhu. Information-theoretic capacity analysis in triangle cellular communication system. In *Proc. IEEE VTC*, pages 1–5, 2006.

[154] The implementation of Ran2Ran NDD in Matlab. `https://github.com/fei-tong/Ran2Ran-NDD`.

[155] Wendi B Heinzelman, Anantha P Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *Wireless Communications, IEEE Transactions on*, 1(4):660–670, 2002.

[156] Dorra Ben Cheikh, Jean-Marc Kelif, Marceau Coupechoux, and Philippe Godlewski. SIR distribution analysis in cellular networks considering the joint impact of path-loss, shadowing and fast fading. *EURASIP Journal on Wireless Communications and Networking*, 2011(1):1–10, 2011.

[157] Xiaoshen Song, Ruonan Zhang, Jianping Pan, and Jiajia Liu. A statistical geometric approach for capacity analysis in two-hop relay communications. In *Proc. IEEE GLOBECOM*, pages 4823–4829, 2013.

[158] Zih-Siang Syu and Chia-Han Lee. Spatial constraints of device-to-device communications. In *Proc. BlackSeaCom*, pages 94–98, 2013.

[159] Daohua Zhu, Jiaheng Wang, A.L. Swindlehurst, and Chunming Zhao. Downlink resource reuse for device-to-device communications underlaying cellular networks. *IEEE Signal Processing Letters*, 21(5):531–534, 2014.

[160] Mobashir Mohammad, XiangFa Guo, and Mun Choon Chan. Oppcast: Exploiting spatial and channel diversity for robust data collection in urban environments. In *Proc. ACM/IEEE IPSN*, pages 1–12, 2016.

[161] Yafeng Wu, Kin Sum Liu, John A Stankovic, Tian He, and Shan Lin. Efficient multichannel communications in wireless sensor networks. *Sensor Networks, ACM Transactions on*, 12(1):3:1–3:23, 2016.

[162] Meng-Shiuan Pan and Yi-Hsun Lee. Fast convergecast for low-duty-cycled multi-channel wireless sensor networks. *Ad Hoc Networks*, 40:1–14, 2016.

[163] Ram Ramanathan, Jason Redi, Cesar Santivanez, David Wiggins, and Stephen Polit. Ad hoc networking with directional antennas: a complete system solution. *IEEE Journal on Selected Areas in Communications*, 23(3):496–506, 2005.