2016

Performance Evaluation of Connectivity and Capacity of Dynamic Spectrum Access Networks

Osama Al-tameemi
University of Central Florida

Performance Evaluation of Connectivity and Capacity of Dynamic Spectrum Access Networks

by

Osama Abbas Hussein Al-Tameemi
B.Sc., Al Nahrain University, Iraq, 2005
M.Sc., Al Nahrain University, Iraq, 2008

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering
in the Department of Electrical and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2016

Major Professor: Mainak Chatterjee

ABSTRACT

Recent measurements on radio spectrum usage have revealed the abundance of under-utilized bands of spectrum that belong to licensed users. This necessitated the paradigm shift from static to dynamic spectrum access (DSA) where secondary networks utilize unused spectrum holes in the licensed bands without causing interference to the licensed user. However, wide scale deployment of these networks have been hindered due to lack of knowledge of expected performance in realistic environments and lack of cost-effective solutions for implementing spectrum database systems. In this dissertation, we address some of the fundamental challenges on how to improve the performance of DSA networks in terms of connectivity and capacity. Apart from showing performance gains via simulation experiments, we designed, implemented, and deployed testbeds that achieve economics of scale.

We start by introducing network connectivity models and show that the well-established disk model does not hold true for interference-limited networks. Thus, we characterize connectivity based on signal to interference and noise ratio (SINR) and show that not all the deployed secondary nodes necessarily contribute towards the network's connectivity. We identify such nodes and show that even-though a node might be communication-visible it can still be connectivity-invisible. The invisibility of such nodes is modeled using the concept of Poisson thinning. The connectivity-visible nodes are combined with the coverage

shrinkage to develop the concept of effective density which is used to characterize the connectivity. Further, we propose three techniques for connectivity maximization. We also show how traditional flooding techniques are not applicable under the SINR model and analyze the underlying causes for that. Moreover, we propose a modified version of probabilistic flooding that uses lower message overhead while accounting for the node outreach and interference. Next, we analyze the connectivity of *multi-channel* distributed networks and show how the invisibility that arises among the secondary nodes results in thinning which we characterize as channel abundance. We also capture the thinning that occurs due to the nodes' interference. We study the effects of interference and channel abundance using Poisson thinning on the formation of a communication link between two nodes and also on the overall connectivity of the secondary network.

As for the capacity, we derive the bounds on the maximum achievable capacity of a randomly deployed secondary network with finite number of nodes in the presence of primary users since finding the exact capacity involves solving an optimization problem that shows in-scalability both in time and search space dimensionality. We speed up the optimization by reducing the optimizer's search space. Next, we characterize the QoS that secondary users can expect. We do so by using vector quantization to partition the QoS space into finite number of regions each of which is represented by one QoS index. We argue that any operating condition of the system can be mapped to one of the pre-computed QoS indices using a simple look-up in $\mathcal{O}\log(N)$ time thus avoiding any cumbersome computation for

QoS evaluation. We implement the QoS space on an 8-bit microcontroller and show how the mathematically intensive operations can be computed in a shorter time.

To demonstrate that there could be low cost solutions that scale, we present and implement an architecture that enables dynamic spectrum access for any type of network ranging from IoT to cellular. The three main components of this architecture are the RSSI sensing network, the DSA server, and the service engine. We use the concept of modular design in these components which allows transparency between them, scalability, and ease of maintenance and upgrade in a plug-n-play manner, without requiring any changes to the other components. Moreover, we provide a blueprint on how to use off-the-shelf commercially available software configurable RF chips to build low cost spectrum sensors. Using testbed experiments, we demonstrate the efficiency of the proposed architecture by comparing its performance to that of a legacy system. We show the benefits in terms of resilience to jamming, channel relinquishment on primary arrival, and best channel determination and allocation. We also show the performance gains in terms of frame error rater and spectral efficiency.

To my family and teachers.

ACKNOWLEDGMENTS

I am grateful to my advisor Dr. Mainak Chatterjee for guiding, supporting and believing in me over the years of my Ph.D studies. Without his experienced mentoring, my dream of earning a Ph.D would not have come true. I would like to express sincere appreciation of my committee members Dr. Mostafa A. Bassiouni, Dr. Lei Wei, Dr. Sumit K. Jha, and Dr. S. Roy Choudhury for serving in my committee. Their constructive feedback and comments have helped me in improving my dissertation. I would like to thank Dr. Kevin Kwait and Dr. Charles Kamhoua from the Air Force Research Laboratory for the collaborative opportunities. I would like also to thank all my colleagues at the NetMoC laboratory and all my friends who have always inspired me while working as a Ph.D. student. In addition, I would like to thank the Department of Electrical and Computer Engineering at the University of Central Florida for offering me with all these wonderful classes which enriched my knowledge along the course of my education. I would like to acknowledge the role of my parents for motivating and supporting my desire of pursuing higher studies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The most vital resource for any wireless application or service (i.e., mobile telephony, TV and radio broadcasts, GPS, maritime navigation, tactical communications, etc.) is the *radio spectrum*. These services operate on specific portions of the electromagnetic spectrum which have been statically allocated for the said purpose. These long-term spectrum allocations and management are usually done under the strict guidance of a governmental agency. In the US, the Federal Communications Commission (FCC) sets the rules and regulations that govern the access to spectrum. These rules have led to allocation of spectrum chunks for specific purposes. Given the limited workable spectrum for any service, there is a fundamental theoretical limit on the achievable capacity as originally shown by Shannon [1].

It can be noted that the spectrum allocation and management have traditionally followed a 'command-and-control' approach and allocated spectrum to specific services under restrictive licenses. The restrictions specify the technologies to be used and the services to be provided, thereby constraining the ability to make use of new technologies and the ability to redistribute the spectrum to higher valued users. In most countries, most frequencies have been completely allocated to specific uses and spectrum appears to be a scarce resource within the current regulatory framework. There have been experimental studies that reveal that spectrum utilization is time and space dependent and that most parts of radio spectrum

are highly underutilized [2]. These limitations along with dis-proportionate and time-varying demand of radio services have motivated a paradigm shift from static spectrum allocation towards a more 'liberalized' notion of dynamic spectrum management in which secondary networks/users (non-licence holders) can 'borrow' idle spectrum from those who hold licensees (i.e., primary networks/users), without causing harmful interference to the latter – a notion commonly referred to as the dynamic spectrum access (DSA) or open spectrum access [3, 4]. Though currently available hardware have limited capability in harnessing under-utilized spectrum, the breakthroughs in cognitive radio technologies [5, 6], empowered by software defined radios (SDR) [7], are poised to promote efficient use of spectrum by adopting this open spectrum approach.

A typical DSA network consists of secondary users who coexist with a network of primary users (spectrum owners). With unregulated deployment of such DSA networks, it is unclear how these networks will perform in terms of assuring quality-of-service (QoS) to the end users. In that regard, we would at least like to know whether the network would even remain connected or not, and if so, what would be the expected capacity. It is worth mentioning that information dissemination, routing, locating resources, and advertisement of services are highly dependent on the network's connectivity. The more the network is connected, the higher are the chances of having these services being available to the secondary users.

Finding solutions to the aforementioned issues is non-trivial since there could be several questions that might arise with regard to connectivity. Some pertinent questions are:

When will the giant component that spans most of the nodes emerge? If it emerges, how big is it? What are the network parameters (including primary density and secondary density) under which such component emerges? For a given network, how can the connectivity be maximized? For a given area, what is the optimal deployment density what would achieve connectivity maximization? What are the effects of multi-channel on the network's connectivity?

While the above mentioned questions are challenges by themselves, what makes them even more difficult is the additional constraints posed by the primary users. Constraints such as the tolerable limits on the signal to interference and noise ratio (SINR) limit the transmission capabilities of secondary users and thus affects their coverage and capacity. For finite networks, finding that exact values for the expected capacity involves solving a non-convex functional constrained optimization problem which can be infeasible to solve in real-time. Several issues arise: How can we find a bound to the maximum capacity of a deployed network in a finite time? Is there way to speed-up the optimization process? Can dimensionality reduction be applied without compromising the solution? Is it possible to achieve a sub-optimal solution in a shorter time? Though in theory, we can find reasonable solutions to these problems, the question remains if such solutions are viable in practice. Given the limited computational capability of power constrained DSA radios, can we get the desired results within a time budget.

In addition to the aforementioned theoretical challenges, there is the considerable sensing overhead in DSA networks which has hindered their deployment. The International

Telecommunications Union (ITU) has shown that sensing is the most expensive part of the DSA system [8]. Moreover, sensing for vacant channels has been a major component for most detection techniques that have been proposed by the Federal Communications Commission (FCC) (in the US) for the realization of DSA systems [9, 10].

To solve this issue and to provide a consistent assay on spectrum vacancies, as well as to have less reliance on sensors for sensing, database driven spectrum allocation and access policies have been mandated [11]. However, even with such approach, having *enough* spectrum sensors to feed the database (and the derived services like the radio environment map) remains a bottleneck from a scalability perspective as such approaches rely on expensive hardware platforms for the sensing tasks [12]. Open questions remain to be answered are: How can we find a solution to the sensing problem? Can we overcome the need for expensive hardware sensing platforms? How can we address the cost-scalability issues? Is it possible to come up with a new design paradigm for a low cost database DSA driven system? Is it possible to incorporate the current available technologies from different vendors under one system? If so, how? Although answers to these questions can be argued and debated, the real challenge is to demonstrate the performance of a database assisted DSA system via real implementation.

1.1 Contributions of this Work

This dissertation addresses some of the fundamental challenges that DSA networks face today. We not only propose novel ways for DSA networks to perform better, we quantify the expected performance for a given network setting. The main contributions of this dissertation are as follows.

1. We characterize the connectivity of a continuum secondary network under the SINR model using percolation theory and stochastic geometry. We show how the well-established condition for continuum percolation does not hold true in the SINR regime. Thus, we find the condition which characterizes the connectivity of such networks. We show how and why not all the deployed secondary nodes necessarily contribute towards the network's connectivity even though they might participate in the communication process. We model the invisibility of such nodes using the concept of Poisson thinning. We combine the thinning with the coverage shrinkage to develop the concept of percolation visible nodes (effective density). Further, we propose three techniques for connectivity maximization.

2. We show how traditional flooding techniques are not applicable under the SINR model and analyze the underlying causes for that. Further more, we identify the source of duplicate transmissions in such techniques. To increase node outreach in interference-limited DSA networks, we propose a modified version of probabilistic flooding that uses lower message overhead without compromising network connectivity.

3. We analyze the connectivity of *multi-channel* distributed networks under the SINR model. The invisibility that arises among the secondary nodes results in thinning which we characterize as channel abundance. We also capture the thinning that arises due to the nodes' interference. We study the effects of interference and channel abundance using Poisson thinning on the formation of a communication link between two nodes and also on the overall connectivity of the secondary network.

4. We derive the bounds on the maximum achievable capacity of a randomly deployed secondary network with finite number of nodes in the presence of primary users since finding the exact capacity involves solving an optimization problem that shows inscalability both in time and search space dimensionality. We speed up the optimization by reducing the optimizers search space. The derived bounds provide insights on the networks maximum and minimum achievable capacities.

5. We characterize the QoS that secondary users can expect in a DSA network in the presence of primaries. We use vector quantization to partition the QoS space into finite number of regions each of which is represented by one QoS index. We argue that any operating condition of the system can be mapped to one of the pre-computed QoS indices using a simple look-up in $\mathcal{O}\log(N)$, time thus avoiding any cumbersome computation for QoS evaluation. We implement the QoS space on an 8-bit microcontroller and show how the mathematically intensive operations can be computed in a short time.

6. We break the cost-scalability barrier of DSA systems by designing a low cost *modular* DSA system architecture consisting of multiple components. Each component has clearly defined input-output functions that allow it to be independently and easily modified in a plug-n-play manner, without requiring any changes to the other components. Moreover, we provide a blueprint on how to use the off-the-shelf commercially available software configurable RF chips to build low cost spectrum sensors. We implement the proposed DSA system architecture and show its performance. Finally, we quantify the gains from having DSA based communications versus legacy communications.

1.2 Organization of the Dissertation

The dissertation is organized as follows. Chapter 2 presents the related work that is relevant to this dissertation. In Chapter 3, the connectivity of DSA networks is analyzed and the corresponding maximization techniques are explained. Chapter 4 discusses the capacity bounds of finite DSA networks and ways to optimize them. Chapter 5 presents the design and implementation of a low cost DSA system. In Chapter 6, the simulation model and results are presented. Conclusions are drawn in Chapter 7.

CHAPTER 2: RELATED WORK

In this chapter, we discuss the literature that is most relevant to this dissertation. We divide the discussion into i) connectivity, ii) capacity of distributed wireless networks, and iii) implementation of DSA Systems

2.1 Connectivity of Distributed Wireless Networks

There is a rich literature with classical results for connectivity of both homogeneous [13, 14, 15] and heterogeneous [16, 17, 18, 19] networks under both the Boolean model [20] and the SINR model [21]. Meester and Roy in [22] were the first to show the percolation condition for continuum networks under the Boolean model is $\lambda > \lambda^c$, where λ is the deployment density and λ^c is some critical density. The authors in [13] studied percolation for scenarios where the nodes interfere each other, where they showed that even under the SINR model, the resulting connectivity is similar to that of a Boolean model given that the interference from other nodes can be sufficiently reduced at each receiving node. Similarly, Vaze in [23] found the receiver threshold for which the condition $\lambda > \lambda^c$ indicates percolation under the SINR model. In [14], continuum percolation was used to obtain the critical transmission range of the asymptotic connectivity of independent identically distributed nodes in the limiting

case. In [15], the k-connectivity of a random graph was linked with the minimum node degree for graphs with high number of nodes. In [16], the k-connectivity of cognitive radio network (CRN) was studied using percolation and asymptotic-connectivity where the critical secondary density was found. In [17], the percolation degree of the secondary users and its relation with the k-disjoint paths was studied in large scale CRNs. In [18], the activity of the primary users was assumed to be dynamic; its effects on the available number of channels for the secondary network was studied using percolation theory in which the critical primary user densities were illustrated. In [19], percolation was used to study the connectivity of heterogeneous networks via identifying connectivity regions which were characterized by a set of primary and secondary networks in which percolation occurs. In [24], black-holes that were caused by nodes failures and their spread properties in CRN was characterized using percolation theory. In [25], it was found that cooperation among heterogeneous secondary networks leads to better connectivity if the cooperating secondary networks percolate while the non-cooperating ones do not percolate.

All the above mentioned works provide useful theoretical insights. However, they did not take into consideration the availability of multi channels for DSA networks and the physical limitation of DSA radios being not able to communicate over a maximum number of channels due to hardware or scanning limitations; thus the connectivity under such scenarios remain unsolved under both the Boolean and SINR models. The percolation conditions under the SINR model is still unknown. Moreover, even the works that studied percolation under the SINR model modified the percolation condition of the Boolean model to make

it applicable to the SINR model (i.e., special cases) rather than explicitly identifying the behavior of connectivity in the SINR regime.

2.1.1 Flooding

Flooding is a popular technique where information from a node is sent to all other nodes in the network. Though easy to implement, flooding causes excessive messaging and communication overhead. Oftentimes, flooding is used to analyze reachability in networks.

In [26], a dynamic source routing protocol for ad hoc networks that uses flooding is presented. In [27], it was shown that the implicit redundancy built in conventional flooding provides resistance against high degree of message losses and node failures as well [27]. In order to reduce the number of re-transmissions, variations of conventional flooding have been proposed [28] like, Time to Live in AODV [29], random walks [30], probabilistic flooding [31], and teeming [32]. Haas et. al. [31] were one of the earliest to notice that flooding of a gossip in a connected graph under the Boolean model exhibits a bimodal behavior, i.e., there was a critical value of the re-boradcast probability (P) above which the gossip spreads to almost all the nodes in the network and below which the gossip dies out. The optimization for P was solved via simulation in [31] and it was shown that the optimized P could reduce the number of broadcasts by 35% when applied to AODV. Similarly, authors in [33] investigated probabilistic flooding in mobile ad-hoc networks, while in [34] the authors bounded the latency of information spread. Probabilistic flooding was also analyzed in [35] for random

networks. A non-uniform value of P that is inversely proportional to the node degree was proposed in [27].

All the previous work considered random wireless networks under the Boolean model. However, flooding and probabilistic flooding were not investigated under the SINR model.

2.2 Capacity of Distributed Wireless Networks

Though the expected performance of traditional wireless networks networks have been well-studied, such studies are still being pursued for DSA networks. In the recent past, there have been several works that discussed the capacity of CRNs in overlay and underlay modes. In [36], the transmission capacity of the hybrid CRN was studied where underlaid and overlaid access approaches were combined to improve the transmission capacity of CRN and it was shown that the hybrid mode yields higher capacity than the underlay or overlay especially when the primary network had light load. In [37], the capacity of the CRN was enhanced using a special receiver and frame structure for spectrum sharing. In [38], the transport capacity of single hop CRN that used Aloha for MAC and nearest neighbor protocol for routing was derived. In [39], the secrecy capacity of CRN was investigated and secure communications in terms of primary and secondary users and the eavesdroppers were highlighted. In [40], the asymptotic capacity of CRN was studied. In [41], the asymptotic capacity of CRNs with static primary users and heterogeneous mobile secondary users were studied. In [42], the capacity limits of cognitive radio (CR) networks with a hybrid relay

scheme in which the primary nodes dominate the spectrum usage, while coordinating with secondary or CR nodes in forwarding packets were investigated. In [43], the Erlang capacity in coordinated cognitive radio networks with VoIP traffic was investigated where an analytical model was developed by employing time-scale decomposition technique and conventional continuous and discrete time Markov chain tools. In [44], the Erlang capacity in coordinated cognitive radio networks with real-time hard delay constraint traffic as function of the rented resources was analytically calculated. In [45], the transmission capacity was increased by using a spectrum sharing scheme with preservation regions to avoid excessive interference from the secondary users to the primary users. In [46], an amplify-and-forward-based cooperative signaling scheme that employs power control to prevent harmful noise propagation was presented and the capacity of the resulting MIMO link was derived. In [47], call admission control algorithms are proposed for slotted ALOHA channel access CRNs where the capacity of the schemes was analyzed.

It is to be noted that the approaches in the above mentioned works are asymptotic in nature and do not provide the capacity for an arbitrary network with finite number of nodes. Moreover, they do not provide any mechanism to improve the optimizer's scalability and execution time which leaves the optimizer's search space dimensionality reduction as an open problem. In addition, they do not investigate the topological effects that result from the node's pairing and the corresponding capacity of the network.

2.2.1 QoS Evaluation

There have been some power control algorithms that have been proposed that address QoS issues in CRNs [48, 49, 40, 50, 51]. In [48], a two-phase mixed distributed/centralized control algorithm is developed for a point to multi point scenario, in which the objective was to maximize the throughput of the CRN with minimal cooperation between the cognitive and the primary devices. In [49], a distributed power control protocol was proposed to enable effective spectrum sharing between primary and secondary users to maximize the aggregate capacity. It also ensured that the interference incurred to the primary was within the interference limit. In [40], the network performance was optimized via performing optimal control on each node, and a formal mathematical model for joint power control, scheduling, and routing was developed. In [50], the optimal power control in CRN was modeled as a concave minimization problem. In [51], a power control algorithm is designed with low implementation complexity through reinforcement learning, which did not require the interference channel and power strategy information among users. From the above it can be noted that all such algorithms are either centralized or distributed in nature, and capable of handling both real time and non-real time traffic. Some also consider different classes of traffic with various QoS requirements.

Although the above mentioned works study power control from different aspects; however none addresses how to evaluate the QoS metrics for the corresponding optimized

power vector. Thus the problem of evaluation on cognitive devices with limited processing power remains open.

2.3 DSA System Implementations

Most prior DSA system implementations either used specialized components (mainly software defined radios) or implemented sensing algorithms that improve the primaries' detection probability [52, 53, 54, 55, 56]. In [52], a real time hardware for spectrum sensing based energy detection with reception diversity was implemented using Stratix II EP2S180 and FPGA boards. In [53], a spectrum sensor for DVB-T signals was implemented using dedicated FPGA and receiver boards with the emphasis being on the signal processing part on the FPGA board. In [54], a cognitive radio device was designed and built with the main focus being small form factor and flexible configurability. In [55], a spectrum sensor was implemented on FPGA board with an adaptive threshold technique to increase the primary's detection probability. In [57], a spectrum sensing network for the UHF TV band from 470-870 MHz was built and its readings were used by a cognitive engine to increase the operational reliability. In [56], a double thresholding primary detection algorithm was implemented on an open wireless research platform (WARP) which showed improvements in the primary's detection probability. In [10], spectrum sensors were built for the 470-870 MHz TV band in which Ethernet enabled Spartan-6 FPGA boards, MicroBlaze boards, and RXIC boards were used. However, none of the above mentioned implementations addressed:

i) the issue for the cost of the deployment, ii) how scalable the systems were and at what price point, and iii) the gains achieved compared to the legacy systems. A part of this dissertation addresses these questions by implementing a low-cost database assisted DSA system that scales.

CHAPTER 3: CONNECTIVITY OF DSA NETWORKS

This chapter studies and characterizes the connectivity of DSA networks under the Boolean and SINR models. We start by deriving the percolation condition for DSA networks under the SINR model and compare it with that under the Boolean model. Next, we use our findings on connectivity and interference to study the node outreach problem in DSA networks. We study the performance of traditional and probabilistic flooding techniques under the SINR model where we show that some nodes are rendered in-usable due to interference. To solve that problem, we developed a modified version of probabilistic flooding which accounts for interference. We compare its performance with the two aforementioned flooding techniques. Finally, we characterize the connectivity of multi-dimensional DSA networks under the Boolean and SINR models. The contents of this chapter appeared in [58, 59, 60, 61, 62].

3.1 Connectivity in DSA Networks

Nodes of a connected network can communicate with each other over one or multiple hops, whereas in a disconnected network there are two or more islands of subnetworks whose nodes cannot communicate with other subnetworks. For any secondary DSA network to provide sustained communication services, it is important that the network remains connected where

radios (nodes) can communicate with each other directly or via intermediate nodes. Thus, knowing the connectivity properties is of utmost importance as they play a major role in determining the network's expected QoS including throughput, reliability, routing, and communication range. We point out that connectivity depends on a number of factors like node density, radio transmission range of nodes, availability of channels, and a node's ability to decode signals in the presence of noise.

The concept of asymptotic connectivity is usually used for studying homogeneous networks where all nodes in the network are connected with high probability when the number of nodes approaches infinity [16]. However, in the presence of primaries, it is not necessary that the secondary nodes remain connected with high probability. even in the limiting case because primary transmissions have a profound impact on secondary transmissions both in time and space. Thus, under such circumstances it becomes necessary to investigate the connectivity of the secondaries which is defined as the existence of an unbounded connected component of its nodes [23].

3.1.1 Connection Models

For a continuum network (in $\mathcal{R}^2$), the connection model plays an important role in determining its connectivity since it defines the rules for establishing edges between the nodes. The two most common connection models for wireless networks are the Boolean and the SINR models.

3.1.1.1 Boolean Model

In the Boolean model, also known as the disc model, the nodes have a fixed range and two nodes are said to be connected if and only if they are within that range (Euclidean distance) of each other [20].

We point out that, this is the most commonly used connection model in studying the connectivity of wireless networks without interference [17, 18, 19]. Bypassing interference can be attributed to interference cancellation or medium access control (MAC) protocol techniques. Although the Boolean model is simple in nature, however it cannot answer if two nodes can communicate (i.e., can decode each other's signal) when there are other transmitting nodes in the vicinity. This is why the SINR model becomes useful.

3.1.1.2 SINR Model

The SINR model is effective where nodes transmit on the same channel at the same time causing interference to each other. An interferer is a transmitter that is in the vicinity of a receiver and is transmitting on the same channel that the receiver is using. Even with such interferers, correct decoding of the intended signal is still possible if the SINR is greater than some pre-defined threshold. A bi-directional link requires the SINR at both receivers/transceivers to be greater than the threshold. Thus, for interference-limited systems, the condition for successful transmission, which also denotes connectivity, from S_j to S_i on a

given channel is:

$$SINR_{j,i} = \frac{\frac{P}{x_{j,i}^{\alpha}}}{\sum\limits_{x_{k,i} \leq r_I \ , \forall k \neq i,j} \frac{\gamma \times P}{x_{k,i}^{\alpha}}} \geq \beta \tag{3.1}$$

where P is the transmission power in Watts of each secondary transceiver, α is the path loss exponent, $x_{j,i}$ is the Euclidean distance between S_i and S_j, γ is the interference cancellation parameter (also known as the processing gain which abstracts the gain from coding, interference cancellation schemes, etc.). Note that, the parameter γ allows the receiver to decode transmissions from more than one transmitter, just like a code division multiple access (CDMA) system.

3.2 Connectivity of Interference Limited DSA Networks

We study the connectivity of DSA networks under the SINR model using percolation theory and investigate how can we maximize connectivity. Let us first discuss some preliminaries of percolation theory.

3.2.1 Percolation Theory Preliminaries

Phenomena such as impact propagation, reachability, and formation of connected clusters in a graph can be best studied using percolation theory as it provides the appropriate tools

to study the connectivity of heterogeneous entities in i) the continuum space and ii) the multi-dimensional lattice $\mathcal{L}^d$, where d is the number of dimensions [22, 63].

Percolation theory deals with the existence and formation of an infinite connected component in a graph [64]. The term percolation on the other hand, refers to the formation of the infinite component which is usually referred to as the infinite cluster. The infinite cluster is usually referred to as C_{max} and is defined as: $|C_{max}| = \infty$, where $|.|$ is the cardinality operator [64]. With no condition on any particular node inclusion, the formation probability of the infinite cluster is given by ψ_p. By Kolmogorov zero-one law, ψ_p is either 0 or 1 [65]. Percolation probability, θ_p, refers to the probability of the formation of the infinite cluster conditioned on including a particular node (which could be the origin). Thus, θ_p also refers to the probability of any node being contained in C_{max}. The terms super and sub-critical are used to indicate whether the network is percolated or not, respectively

There are two types of percolation i) discrete percolation, and ii) continuum percolation. The former studies phase transitions in the discrete grid (lattice), while the latter studies phase transitions in the continuum space [63]. Discrete percolation has been extensively studied [63, 66, 67, 68]. However due to the continuum nature of the wireless networks, we use continuum percolation for our analysis. As for continuum percolation, Meester and Roy in [22] were the first to characterize the percolation condition under the Boolean model which is $\lambda > \lambda^c$, where λ is the deployment density and λ^c is the critical density which is defined as the minimum deployment density for which the network percolates for the first time.

Finally, we point out that, in this work, we use the words 'percolation' and 'connectivity' synonymously to indicate the formation of the infinite component C_{max} i.e., percolation implies $\theta_p = P(|C_{max}| = \infty) > 0$ and $\psi_p = 1$.

To summarize, in our context, we use percolation theory to characterize the network parameters (primary and secondary densities, coverage radius, percolation probability) that achieve the phase transition i.e., formation of an infinite connected secondary network from multiple isolated smaller networks.

3.2.2 System Model

We consider a secondary DSA network co-existing with primary users. The secondary users (transceivers) are distributed according to a Poisson point process of density λ_s nodes per unit area. Each secondary user has maximum coverage radius r_s (achieved under minimal interference) and interference radius r_I, where $r_I > r_s$. r_I is the maximum distance from which the transceiver can detect an interference from another secondary transmitter. Each receiver (transceiver) has a receive threshold β and noise cancellation factor γ. γ is also known as the processing gain and it abstracts the gains from coding, interference cancellation techniques etc. Note that, γ allows the transceiver to decode the transmissions from the neighboring transceivers. A neighbor of a node is any other node that lies within a distance of r_s from itself.

The primary users (transceivers) are also distributed following the Poisson point process with density λ_p users per unit area. Each primary user has a protection radius of r_p and transmit power P_p. In this work as in [19, 69, 70] we use zero-interference tolerance inside r_p (i.e., secondaries within r_p evict on primary presence). Also, as in the percolation literature [16, 19, 69, 70] in order to keep the analysis and the derived results generic, no power control is used for the secondary transceivers, i.e., all the secondary nodes transmit with the power P_s as long as they are not within r_p from any primary user(s). n denotes the number of secondary users.

3.2.3 Percolation Under SINR Model

Till date, the condition of percolation for continuum networks under the Boolean model, $\lambda > \lambda^c$, has been considered the closest approximation in characterizing percolation. We will prove that this condition does not hold true under interference i.e., it does not necessarily indicate percolation. The following lemma is not limited to DSA networks only as it governs the percolation for any wireless network in the SINR regime by simply assuming no primary presence.

Lemma 3.1. *$\lambda > \lambda^c$ does not apply for continuum networks under the SINR model for a fixed $\gamma \in \mathcal{R}$.*

Proof: The proof is done by contradiction. To keep the proof generic, we assume that there are no primaries.

Assume $\lambda > \lambda^c$ holds for a network with a given (fixed) γ under the SINR model. This means for a constant $k \geq 1$, a density of $k\lambda$ also percolates. Now as k increases, so does the number of neighbors each receiver has. Eventually for some value of k, the receiver's neighbors will generate more interference than what γ can tolerate. This makes the ratio of the numerator of Eqn. (3.1) to the denominator less than β which renders the receiver interfered. As k is increased, more receivers will be interfered. For $k \to \infty$ all the nodes are interfered, thus there are no links in the network. However, the condition $\lambda^c < k\lambda$ is still satisfied, which indicates percolation, thus a contradiction.□

In search for the percolation condition of the secondary network in the SINR regime, we observe that the percolation condition of the Boolean model ($\lambda_s > \lambda_s^c$) considers *all* the deployed nodes as possible candidates that aid the percolation process. In addition to that, it also assumes all the nodes have coverage radius of r_s. We point out that, under such model and due to interference i) some nodes will be invisible to percolation, and ii) the nodes will have their coverage radius reduced. This drives us to i) discount from the λ_s nodes per unit area the ones that are invisible to the percolation process, ii) map the coverage area of the percolation-visible nodes (along with their reduced coverage) to an equivalent density in the Boolean model with a fixed coverage radius for each node. We refer to that equivalent density as the percolation-effective density λ_{eff}. Only then, we can use $\lambda_{eff} > \lambda_s^c$ as the test for percolation since λ_{eff} meets the criteria of the classical Boolean model.

3.2.3.1 Reduced Coverage

Recall that, r_s denotes the maximum coverage radius with no interference. However, due to interference, the maximum coverage radius of each node shrinks as more interferes/neighbors are added. For uniformly deployed neighbors, the probability of having an interferer within a distance x from a deployed node is $f_X(x) = 2x/r_I$ for $0 \leq x \leq r_I$ with $E[f_X(x)] = \frac{2}{3}r_I$. Thus the average coverage radius $\bar{r}_s(N)$ for a node with N secondary interferes is:

$$\bar{r}_s(N) = \frac{\frac{2}{3}r_I}{(N\beta\gamma)^{1/\alpha}} \tag{3.2}$$

The above equation follows directly from Eqn (3.1).

From a single node's perspective, this shrinkage will effect its coverage area. From a percolation's perspective, this will reduce the total coverage area, which is equivalent to reducing the number of deployed nodes. We use the scaling property to reflect this effect on the percolation process. Scaling property states that, percolation of a network with radius r_1 and density λ_s is equivalent to a network whose radius is r_2 and density $\lambda_s \times (r_1/r_2)^2$ [22].

3.2.3.2 Node Invisibility

Now, we develop the basis of classifying a node as *invisible*, which is defined as a physically deployed node that is invisible (in-capable of participating) to the percolation process. We characterize three classes of invisible nodes.

1. **Dominated Nodes:** A 1-dominated node is a node which is prevented from establishing a successful communication with any other neighbor(s) due to the transmission of its closest neighbor. For example, node i (shown in Fig. 3.1(a)) is 1-dominated by j (i's closest neighbor). This implies $SINR_{j,i} \geq \beta$, and $SINR_{j',i} < \beta$ for any other neighbor j'. $SINR_{j',i} < \beta$ is because of the interference from j which is the dominant interferer for any transmission destined to i.

   The concept of dominated nodes exists only from percolation perspective. From a communication perspective, a 1-dominated node is not *interfered* since it is communicating with its closest neighbor successfully.

   In the same manner, we define a 2-dominated node as a node which is dominated by two of its closest neighbors i.e., i is 2-dominated by its closest neighbors j and $\hat{j}$ if $SINR_{j,i} \geq \beta$ and $SINR_{\hat{j},i} \geq \beta$ while $SINR_{j',i} < \beta$ for all the other neighbors due to the interference received from j and $\hat{j}$. Fig. 3.1(b) illustrates a 2-dominated node.

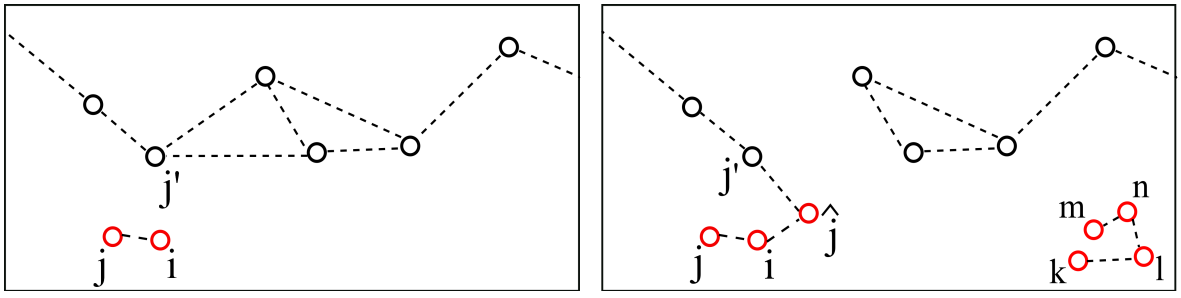
   In [65], it was shown that in order for the network to percolate in the continuum domain, each node needs strictly more than 2 connections i.e., $K > 2$, where K is the number of connections. Computer simulations showed that the lowest value K can be 3 [65]. On this basis, we consider any node which is 1- or 2-dominated as invisible to the percolation process.

2. **Interfered Nodes:** These are the nodes which cannot even sustain at least one connection i.e., $SINR < \beta$ (for all neighbors). This is due to i) interference by surrounding secondary and primary users. or ii) the secondary node being located within

the eviction region of the primary user thus cannot transmit due to the zero-interference tolerance. Having $SINR < \beta$ means that the node is obsolete from a communication perspective which makes it invisible from a percolation's perspective as well.

We point out that, from a communication perspective an interfered node is not the same as a dominated node since the first has no connections while the latter has at least one connection. However, from percolation's perspective both of them are *invisible* to the formation of the infinite cluster.

3. **Nodes within Interference-Isolated Components:** These are nodes that are contained in an interference-isolated component i.e., within 3 or more nodes; however the interference and dominance by the component's nodes prevent them from connecting with other components. Nodes contained in such components are rendered invisible from a percolation's perspective as well. Nodes k, l, m and n in Fig. 3.1(b) form an interference-isolated component.



(a) (b)

Figure 3.1 (a) i is 1-dominated by j, (b) i is 2-dominated by j and $\hat{j}$. The figures also show that j-i in (a) and m-n in (b) participate in the communication process but not in the percolation process.

From the above three cases, it is obvious that under the SINR model not all nodes contribute towards the percolation process. This effectively lowers the deployment density from λ_s to $\lambda_s(1-P_{thin}^{wp})$ where P_{thin}^{wp} is the thinning probability. We use the superscript wp to explicitly denote the probabilities with the primaries (i.e., primaries are present). Later, we find the same probabilities without the primaries where we drop the superscript. For now, we have:

$$P_{thin}^{wp} = P_{d-1}^{wp} + P_{d-2}^{wp} + P_{inter}^{wp} + P_{iso}^{wp} \tag{3.3}$$

where P_{d-1}^{wp}, P_{d-2}^{wp}, P_{inter}^{wp} and P_{iso}^{wp} denote respectively the probabilities of being 1-dominated, 2-dominated, interfered and isolated in presence of primaries. We point out that all the probabilities in the equation above are functions of λ_s, λ_p, γ, β, r_s, r_I, r_p, and α. Thus *we propose not to use* $\lambda_s > \lambda_s^c$ when testing for percolation in the continuum domain under the SINR model, *but instead consider only the percolation-visible nodes* with their coverage radius scaled to r_s. Thus:

$$\lambda_{eff} > \lambda_s^c \tag{3.4}$$

$$\lambda_{eff} = \lambda_s(1 - P_{thin}^{wp}) \times \left(\bar{r}_s(N)/r_s\right)^2 \tag{3.5}$$

Next, we proceed to analytically characterize all the aforementioned probabilities.

3.2.4 Characterization of Thinning Probability

Due to the complexity involved in calculating P_{thin}^{wp} and in order to make the analysis and the proposed concepts easier to follow, we start with $\lambda_p = 0$ i.e., no primaries. Such setting exploits percolation in terms of secondary to secondary interference. Once illustrated, we generalize our approach to account for $\lambda_p > 0$ i.e., with primaries.

3.2.4.1 Primaries Absent ($\lambda_p = 0$)

1) P_{d-x} is the probability of being dominated by x closest neighbor(s). To calculate such probability for a node (say i), we iterate through all the possible combinations as follows:

$$P(\text{node } i \text{ is dominated by } x \text{ closest neighbors}) = \sum_{K=x}^{4\lambda_s} \int_{r_1=0}^{r_I} \int_{r_2=0}^{r_I} \cdots \int_{r_K=0}^{r_I} \mathrm{F(x, r_1, r_2, \cdot\cdot, \cdot, r_K)} \times \prod_{z=1}^{K} \frac{2d_{z,i}}{r_I^2} \times \frac{(\mathrm{A\lambda_s})^{\mathrm{K}}\ \mathrm{e}^{-A\lambda_s}}{K!} \mathrm{d}r_1\, \mathrm{d}r_2\, \cdots\, \mathrm{d}r_K \tag{3.6}$$

where A is πr_I^2, r_j is a shortcut for $d_{j,i}$. Note, theoretically the upper bound of λ_s is ∞ as it is Poisson distributed. For better tractability, we bound its maximum value to $4\lambda_s$ which ensures that at least 99.33% of the cases are accounted for as $\sum_{k=0}^{4\lambda_s} \lambda_s^k e^{-\lambda_s}/k! \geq 0.9933$ for $\lambda_s \geq 5$.

$$F(x, r_1, r_2, \cdots, r_K) = \begin{cases} 1 & \text{if } \sum_{j=1}^{K} G_j = x \text{ or } K = x = 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

$$G_j(r_1, \cdots, r_K) = \begin{cases} 1 & \text{if } \dfrac{\frac{P_s}{r_j^\alpha}}{\sum\limits_{\substack{\forall j' \text{ S.T.} \\ r_{j'} \leq r_I, j' \neq j}} \dfrac{\gamma P_s}{r_{j'}^\alpha}} \geq \beta \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

Eqn. (3.6) uses the law of total probability to iterate through all the possible combinations of distances for node i and its neighbors. It tests when node i is dominated by x closest neighbors out of all the available ones. This is why summation is from $K = x$ to $4\lambda_s$. The individual integrations account for the locations of the neighbors one by one, while $F(\cdot)$ and $G(\cdot)$ are the indicator functions.

By setting $x = 1$, we get P_{d-1}; setting $x = 2$ results in P_{d-2}. Continuing in the same fashion, setting $x = M$ results in P_{d-M} (i.e., dominated by M neighbors).

2) P_{inter} is the probability when a node is interfered at all times. Such a node can be viewed as a node which is dominated by 0 neighbors. Since equivalent events have equivalent probabilities, this results in $P_{inter} = P(\textit{dominated by } 0 \textit{ nodes})$ thus:

$$P_{inter} = P_{d-0}. \tag{3.9}$$

Fig. 3.2 illustrates P_{d-0}, P_{d-1} and P_{d-2} for a Poisson distributed secondary network in an area of 300×300 with density λ_s and $\gamma = 0.5$ while Fig. 3.3 illustrates the resultant reduction in coverage radius due to interference. All the analytical plots in this work are evaluated for $r_s = 25$, $r_I = 35$, $\beta = 1$, and $\alpha = 2$. All the equations in this work are numerically evaluated

using the Monte-Carlo method. The results from Monte-Carlo can be smoothened by using number of iterations with high confidence interval.

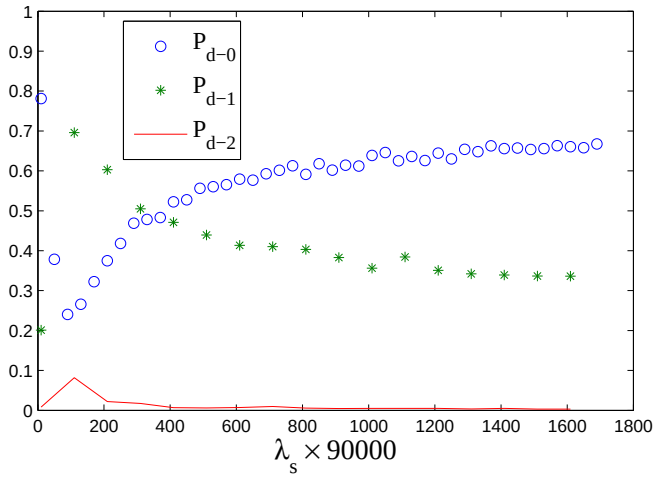


Figure 3.2 P_{d-0}, P_{d-1} and P_{d-2} for $\gamma = 0.5$

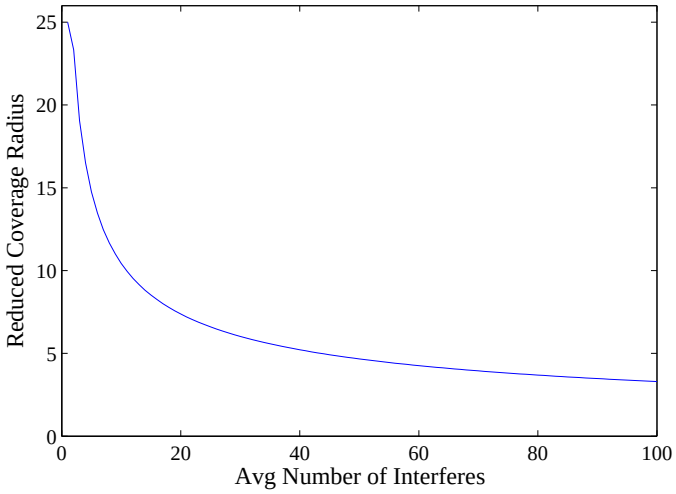
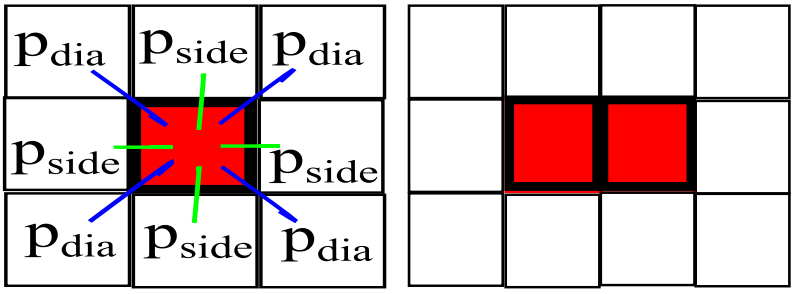


Figure 3.3 Shrinkage in coverage radius versus number of interferes for $\gamma = 0.5$.

3) P_{iso} is the probability that a node is contained in an interference-isolated component. The calculation of this probability is more involved. Since combinatorial counting algorithms in square lattice have been well-studied [71], we consider squares of side r_s. For each such square, we define 8 neighboring squares– 4 on each side and 4 along the corners (or, diagonals). We define the *no-crossing probability* as the probability of having no successful communication between nodes in the neighboring squares due to interference. Two types of no-crossing probabilities arise: p_{side} for no side crossing and p_{dia} for no diagonal crossing.

Fig 3.4(a) shows the no-crossing probabilities for an isolated component that is confined in one square. Similarly, Fig 3.4(b) shows the no-crossing probabilities for an isolated component that is confined in two adjoining squares. For three adjoining squares, there could be two *distinct* different orientations as shown in Fig 3.5. We consider isolated components that reside in up-to 4 squares. With four squares, there are 5 distinct orientations which are shown in Fig. 3.6. Each distinct orientation can occur in multiple ways (i.e., with rotations)– the exact number of which can be found by combinatorial enumeration.



(a) (b)

Figure 3.4 Illustration for p_{side} and p_{dia}. a) Isolated component within 1 square; b) Isolated component within 2 squares.

We find no-side cross probability as:

$$p_{side} = 1 - \sum_{n_1=0}^{4\lambda_s} \sum_{n_2=0}^{4\lambda_s} \int_{x_{11}=0}^{r_I} \cdots \int_{x_{1n_1}=0}^{r_I} \int_{y_{11}=0}^{r_I} \cdots \int_{y_{1n_1}=0}^{r_I} \times$$
$$\int_{x_{21}=0}^{r_I} \cdots \int_{x_{2n_2}=0}^{r_I} \int_{y_{21}=r_I}^{2r_I} \cdots \int_{y_{2n_2}=r_I}^{2r_I} \mathrm{I}(\bar{\mathrm{x}}, \bar{\mathrm{y}}) \times \frac{(\mathrm{A}\lambda_\mathrm{s})^{\mathrm{n}_1+\mathrm{n}_2}\ \mathrm{e}^{-2A\lambda_s}}{n_2! \times n_1! \times r_I^{n_1+n_2}} \mathrm{d}x_{11}\, \mathrm{d}y_{11}\ \cdots \mathrm{d}x_{2n_2}\, \mathrm{d}y_{2n_2} \tag{3.10}$$

where $\bar{x} = \{x_{11}, \cdots, x_{1n_1}, x_{21}, \cdots, x_{2n_2}\}$ $\bar{y} = \{y_{11}, \cdots, y_{1n_1}, y_{21}, \cdots, y_{2n_2}\}$

$$I(\bar{x}, \bar{y}) = \begin{cases} 1 & G(\bar{x}, \bar{y}) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

$$G(\bar{x}, \bar{y}) = \sum_{i=0}^{n_1} \sum_{j=0}^{n2} F(i, j, \bar{x}, \bar{y}) \tag{3.12}$$

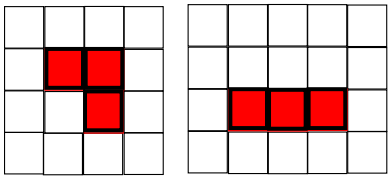


Figure 3.5 Isolated component within 3 squares.

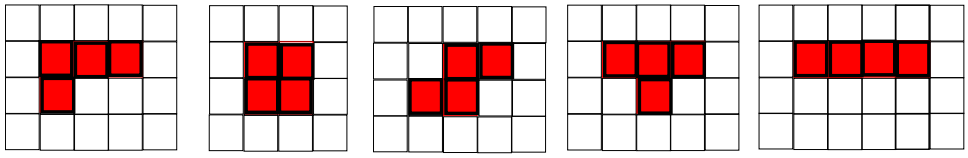


Figure 3.6 Isolated component within 4 squares.

$$F(i,j,\bar{x},\bar{y}) = \begin{cases} 1 & \text{if } SINR_{i,j} \geq \beta \text{ and } SINR_{j,i} \geq \beta \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

Similarly p_{dia} can be found using Eqn. (3.10) with all the x_{2i} terms integrated from r_I to $2r_I$. Fig. 3.7 illustrates p_{dia}.

With p_{side} and p_{dia} known, we use combinatorial counting arguments to find the number and size of closed paths (which consist of non-crossable squares) required to surround/isolate a component of size x-squares. We limit our treatment to isolated components of size up to 4-squares since larger components require longer paths of closed squares to surround them. This also concurs with the simulation results that show isolation of larger components occur with much smaller probability as compared to the probability of surrounding components of 1 up to 4 squares.

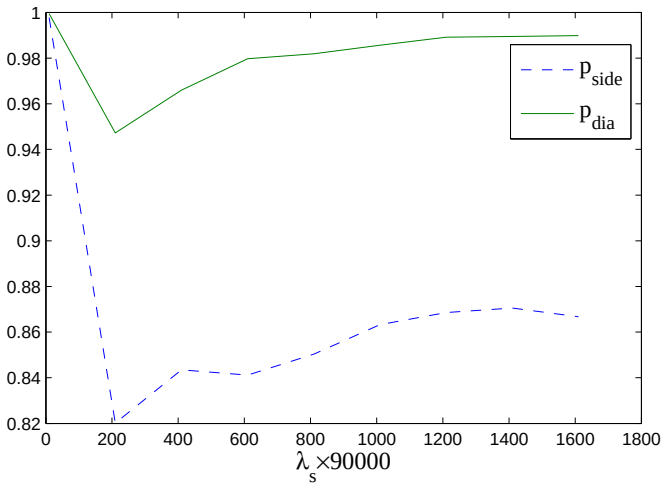


Figure 3.7 λ_s versus p_{side} and p_{dia} for $\gamma = 0.5$, $\lambda_p = 0$.

The isolation probability for the configuration shown in Fig. 3.4(a) (component within a single square) is $p_{side}^4 \times p_{dia}^4$ i.e., all the 8 neighbors (4 on side and 4 on diagonal) has to be closed. For a configuration shown in Fig. 3.4(b) (component within two squares) the isolation probability is $p_{side}^6 \times p_{dia}^4$ (all the 6 side neighbors has to be closed as well as all the 4 diagonal neighbors). Now, the 2 squares could have a different orientation (say, vertical). Accounting for all possible orientations with 2 squares, the isolation probability is $2p_{side}^6 \times p_{dia}^4$. With 3 squares, the isolation probability is $2p_{side}^8 \times p_{dia}^4 + 4p_{side}^7 \times p_{dia}^5$. The two terms are due to the two different configurations (one linear and one L-shaped). Similarly, the isolation probability with 4 squares would contain 5 terms due to the 5 unique configurations. Accounting for all the possible orientations for isolated components with 4 squares leads to an isolation probability of $p_{side}^8 \times p_{dia}^4 + p_{side}^{10} \times p_{dia}^4 + 4p_{side}^8 \times p_{dia}^6 + 4p_{side}^8 \times p_{dia}^6 + 4p_{side}^9 \times p_{dia}^5$. By adding all the possible orientations with their corresponding probabilities for components of sizes 1, 2, 3, and 4, we get:

$$P_{iso} = \sum_{i=1}^{i=4} \text{Prob}\{\text{isolated component within } i \text{ squares}\} \tag{3.14}$$

We acknowledge this approach yields an approximative value for P_{iso} since calculating the exact value is very cumbersome. With p_{side}, p_{dia}, P_{d-0}, P_{d-1}, and P_{d-2} known we can evaluate P_{thin} using Eqn. (3.3) which is shown in Fig. 3.8.

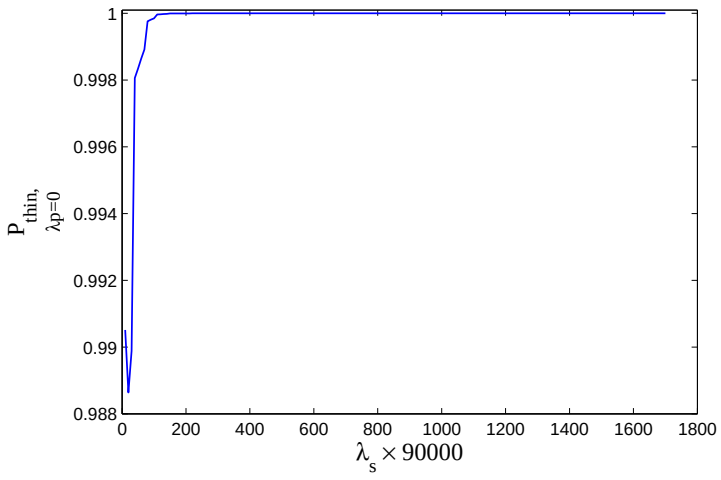


Figure 3.8 λ_s versus total thinning probability for $\gamma = 0.5$, $\lambda_p = 0$.

3.2.4.2 Primaries Present

As mentioned earlier, with primaries present, some secondary users will be evicted (due to primary interference policy) while others will have their SINR lowered due to interference from the primary user(s). Another implication is that, every secondary user has to be checked whether it is evicted or not before being considered as an interferer to other secondary nodes or even being considered as a possible candidate to the percolation process. Finally, interference from the primary users, has to be accounted for when calculating P_{d-0}^{wp}, P_{d-1}^{wp}, P_{d-2}^{wp} and P_{iso}^{wp}. Capturing and reflecting those effects in P_{thin}^{wp}, P_{d-x}^{wp}, P_{inter}^{wp}, and P_{iso}^{wp} constitutes the thinning effects of the primary users. Once again, we use the law of total probability to iterate through all the possible combinations of distances, number of neighbors but this time we also account for the distances and locations (between 0 and 2π) of all the M primary

users as well. This results in:

$$P(\text{node } i \text{ is dominated by } x \text{ closest neighbors})^{wp} = \sum_{M=1}^{4\lambda_p} \int_{R_1=0}^{R_1=r_I+r_p} \int_{\theta_1=0}^{2\pi} \cdots \int_{R_M=0}^{R_1=r_I+r_p} \int_{\theta_M=0}^{2\pi} \sum_{K=x}^{4\lambda_s} \int_{r_1=0}^{r_I} \int_{r_2=0}^{r_I} \cdots \int_{r_K=0}^{r_I} \mathrm{F(x, r_1, \cdot\cdot, r_K, R_1, \cdot\cdot, R_M, \theta_1, \cdot\cdot, \theta_M)} \times \prod_{z'=1}^{M} \frac{2d_{z',i}}{(r_I+r_p)^2} \times \frac{(\mathrm{A_p}\lambda_\mathrm{p})^\mathrm{M}\ \mathrm{e}^{-A\lambda_p}}{M!} \times \prod_{z=1}^{K} \frac{2d_{z,i}}{r_I^2} \times \frac{(\mathrm{A}\lambda_\mathrm{s})^\mathrm{K}\ \mathrm{e}^{-A\lambda_s}}{K!} \mathrm{d}r_1\, \mathrm{d}r_2\, \cdots\, \mathrm{d}r_K \mathrm{d}R_1\, \mathrm{d}R_2\, \cdots\, \mathrm{d}r_M \mathrm{d}\theta_1\, \mathrm{d}\theta_2\, \cdots\, \mathrm{d}\theta_M \quad (3.15)$$

where $A_p = (r_p + r_I)^2\pi$, $\mathrm{F(x, r_1, \cdot\cdot, r_K, R_1, \cdot\cdot, R_M, \theta_1, \cdot\cdot, \theta_M)}$ is similar to Eqn. (3.24), and $\mathrm{G(x, r_1, \cdot\cdot, r_K, R_1, \cdot\cdot, R_M, \theta_1, \cdot\cdot, \theta_M)}$ is similar to Eqn. (3.25) with Eqn. (3.1) used to test for successful communication. As for the no-cross probability:

$$p_{side}^{wp} = 1 - \sum_{M=1}^{4\lambda_p} \int_{R_1=0}^{4r_I+r_p} \cdots \int_{R_M=0}^{4r_I+r_p} \cdots \int_{\theta_1=0}^{2\pi} \int_{\theta_M=0}^{2\pi} \sum_{n_1=0}^{4\lambda_s} \sum_{n_2=0}^{4\lambda_s} \int_{x_{11}=0}^{r_I} \cdots \int_{x_{1n_1}=0}^{r_I} \int_{y_{11}=0}^{r_I} \cdots \int_{y_{1n_1}=0}^{r_I} \int_{x_{21}=0}^{r_I} \cdots \int_{x_{2n_2}=0}^{r_I} \int_{y_{21}=r_I}^{2r_I} \cdots \int_{y_{2n_2}=r_I}^{2r_I} \mathrm{I(\bar{x}, \bar{y}, \bar{R}, \bar{\theta})} \frac{(\mathrm{A}\lambda_\mathrm{s})^{\mathrm{n_1+n_2}}\ \mathrm{e}^{-2A\lambda_s}}{n_2! \times n_1! \times r_I^{n_1+n_2}} \times \frac{(\mathrm{A_p}\lambda_\mathrm{p})^\mathrm{M}\ \mathrm{e}^{-A_p\lambda_p}}{M!(r_p)^M(2\pi)^M} \mathrm{d}x_{11}\, \mathrm{d}y_{11}\, \mathrm{d}R_1\, \mathrm{d}\theta_1\, \cdots \mathrm{d}x_{2n_2}\, \mathrm{d}y_{2n_2}\, \mathrm{d}R_M\, \mathrm{d}\theta_M \quad (3.16)$$

where $\bar{R} = \{R_1, \cdots, R_M\}$, $\bar{\theta} = \{\theta_1, \cdots, \theta_M\}$

$$I(\bar{x}, \bar{y}, \bar{R}, \bar{\theta}) = \begin{cases} 1 & G(\bar{x}, \bar{y}, \bar{R}, \bar{\theta}) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.17)$$

$$G(\bar{x}, \bar{y}, \bar{R}, \bar{\theta}) = \sum_{i=0}^{n_1} \sum_{j=0}^{n2} F(i, j, \bar{x}, \bar{y}, \bar{R}, \bar{\theta}) \tag{3.18}$$

$$F(i, j, \bar{x}, \bar{y}, \bar{R}, \bar{\theta}) = \begin{cases} 1 & if SINR_{i,j} \geq \beta \text{ and } SINR_{j,i} \geq \beta \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

3.3 Maximizing Connectivity

In this section, we apply our findings on effective density to maximize connectivity of DSA networks under the SINR model. To do so, we use three techniques: i) optimal deployment density: we find λ_s that results in the maximum effective density, ii) optimal 'receive-only' ratio: we find the fraction of nodes that would be used in the 'receive-only' mode, and iii) optimal TDMA slotting: we find the optimal number of TDMA slots that is required for a subset of nodes to be active in each time slot. To the best of our knowledge, this is the first work that uses these three techniques to maximize connectivity of interference-limited DSA networks. Based on these three optimization techniques, we propose a medium access control (MAC) protocol for un-coordinated secondary nodes in DSA networks.

3.3.1 Optimal Deployment Density

Unlike the Boolean model where increasing λ_s always increases the connectivity; under the SINR model, increasing λ_s can either increase or decrease the connectivity by virtue of the thinning probabilities as was shown in Figs. 3.2, 3.7 and 3.8. It is to be emphasized that under the SINR model, there is no single function that can characterize connectivity in terms of the deployment density (λ_s). Thus, there is no straight forward objective function that could be optimized.

We approach this problem of connectivity maximization from a different perspective, i.e., from a percolation's point of view. We follow the known results from the Boolean model where connectivity increases with the number of deployed nodes per unit area, which is the same as the density of the percolation visible nodes. In other words, instead of increasing λ_s arbitrarily, we use such a λ_s that maximizes the number of percolation visible nodes (i.e., we seek to maximize λ_{eff}). We take such an approach since λ_{eff} as represented in Eqn. (3.5) can be easily optimized using standard optimization techniques. Thus, the connectivity maximization problem boils down to finding the optimal deployment density, represented by λ_s^{opt} as:

$$\lambda_s^{opt} = \arg\max_{\lambda_s} \lambda_s(1 - P_{thin}^{wp}) \times \left(\bar{r}_s(N)/r_s\right)^2 \tag{3.20}$$

3.3.2 Optimal 'Receive-only' Ratio

Under the SINR model, if the nodes were to transmit at the same time, the resulting interference might hamper the connectivity and reduce the network's coverage. Thus, it makes sense not to allow a certain fraction of nodes to transmit. Thus, we allow a fraction τ of the nodes to remain in receive-only mode. Thus, there are $\lambda_s \times (1-\tau)$ simultaneously transmitting nodes and $\lambda_s \times \tau$ nodes in receive-only mode. Choosing the subset of nodes that need to be in receive-only-mode can be done in two ways.

1. **Individual selection:** In this approach, each node is inspected with respect to all neighboring nodes and beyond if it could be allowed to transmit. This resembles the NP-complete graph coloring problem which is beyond the scope of this work. Moreover, this approach requires some form of communication between the nodes which makes it more complex to implement.

2. **Random selection:** In this approach, a node independently and randomly decides to be in the receive-only mode with probability τ. This distributed approach requires no cooperation between the nodes which makes it easier to implement. We define the optimal receive ratio, τ^{opt}, as the optimal τ that leads to the highest connectivity for a given λ_s.

   Similar to finding the optimal deployment density where the connectivity was maximized at λ_s^{opt}, we propose to reduce the number of transmitting nodes from λ_s to λ_s^{opt}

using:

$$\tau^{opt} = \frac{\lambda_s - \lambda_s^{opt}}{\lambda_s} \qquad \forall \;\; \lambda_s > \lambda_s^{opt} \tag{3.21}$$

Thus, the expected number of receive-only and transmitting nodes are $\tau^{opt} \times \lambda_s$ and λ_s^{opt}, respectively.

We point out that, the concept of putting nodes to receive-only or sleep is also encountered in wireless sensor networks [72]. In that context, τ refers to the fraction of the nodes that are put to sleep with the goal to minimize energy consumption while retaining maximum connectivity. Note that, using τ^{opt} will yield the best connectivity for all values of τ; however it will not be optimal which can only be obtained via individual selection.

3.3.3 Optimal TDMA slotting

Although putting some of the nodes into receive-only mode sounds appealing for networks with $\lambda > \lambda_s^{opt}$, the question remains when does each node gets a chance to transmit, given the fact that only λ_s^{opt} nodes per unit area will transmit simultaneously.

Time Division Multiple Access (TDMA) is a popular allocation scheme where nodes are granted channel access on periodic and timely basis [73, 74, 75]. Essentially, it seeks to to find the number of subsets of nodes that can transmit simultaneously, each subset transmits on a particular time slot. The subsets transmit in a round-robin fashion– thus,

the number of time slots required is equal to the number of subsets. Again, there could be two approaches.

1. **Individual allocation:** This is similar to the individual selection method as was discussed in section 3.3.2 and is beyond the scope of this work.

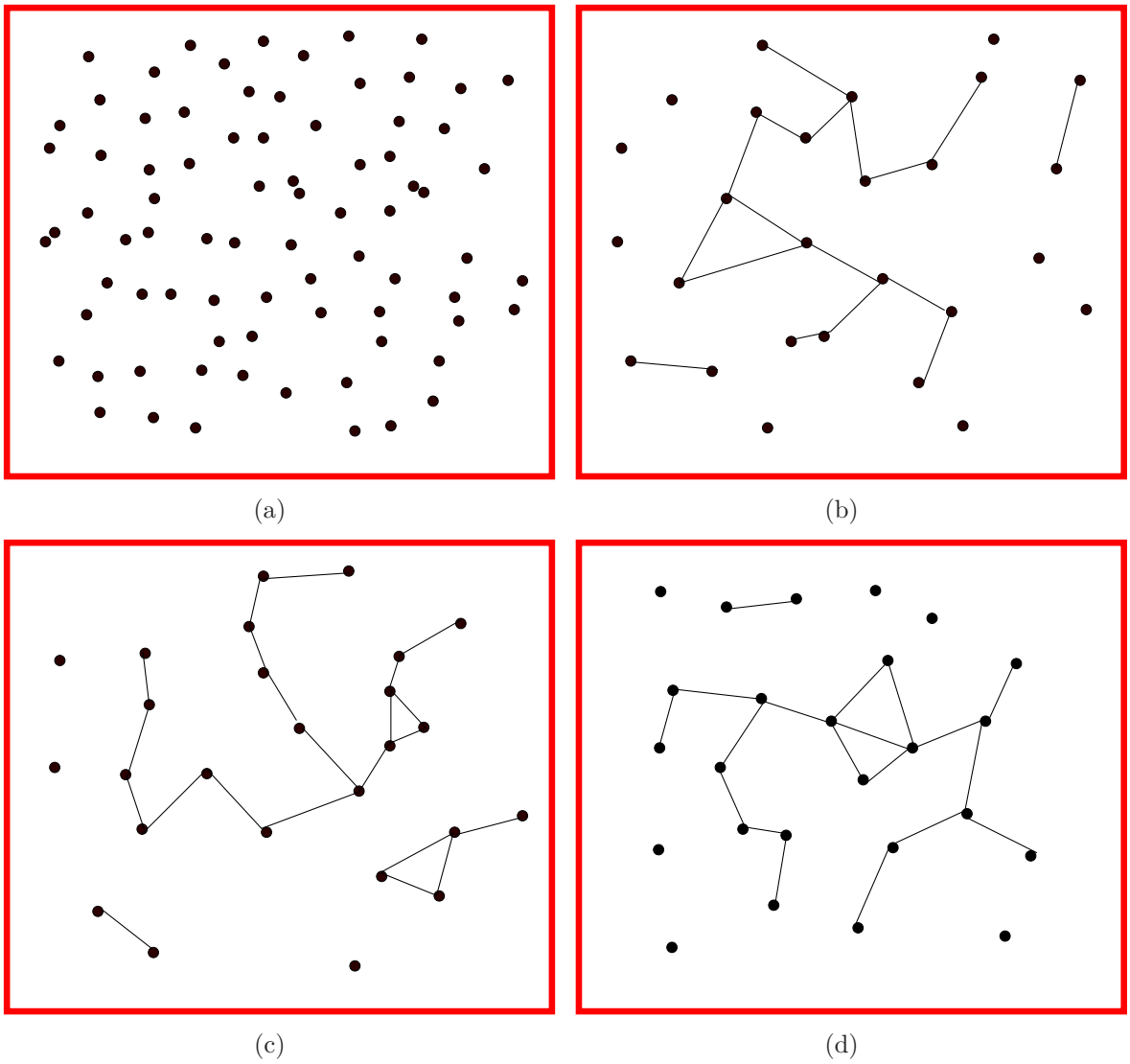


Figure 3.9 (a) A network with a total of 80 nodes. (b) The network during first time slot. (c) The network during second time slot. (d) The network during the third time slot.

2. **Random allocation:** Here, the objective of a node is to independently and randomly select a time slot to transmit. Nodes that select the same time slot will end up transmitting simultaneously on the same time slots in every super-frame as shown in Fig. 3.10.

   Thus, the ones that transmit on slot 1 are subset 1 and so on. Hence, the subset formation is a by-product of random time slot selection. This is similar to the *Poisson Blinking Model* [34] where each node alternates between full duplex transmit-receive-mode and receive-only mode. An illustrative example is shown in Fig. 3.9(a) where a network with 80 nodes has been divided into 3 subsets of 25, 28, and 27. During time slot 1, nodes belonging to subset 1 will transmit resulting in the connected network shown by Fig. 3.9(b). Similarly, in slots 2 and 3, we get the networks as shown in Figs. 3.9(c) and 3.9(d).

   The question is: how many time slots make up a super-frame, i.e., t^{opt}? We recall our finding from sections 3.3.1 and 3.3.2 where we found that connectivity can be maximized by reducing the amount of interference which can be achieved by reducing the number of transmitting nodes such that the average number of transmitting nodes is λ_s^{opt} node per unit area. We extend the same concept as follows. Instead of having λ_s nodes per unit area trying to transmit simultaneously, we will have on average λ_s^{opt} of them transmitting on each time slot. This means to divide the total number of deployed nodes into subsets where on average each group will have λ_s^{opt} nodes per unit

area. This leads to:

$$t^{opt} = \left\lfloor \frac{\lambda_s}{\lambda_s^{opt}} \right\rceil \qquad \forall \ \ \lambda_s > \lambda_s^{opt} \tag{3.22}$$

where $\lfloor \rceil$ denotes the nearest integer.

We point out that, although the connectivity that results with t^{opt} is the best connectivity compared to the ones obtained with the other values of t, the connectivity obtained under t^{opt} is still suboptimal as the allocation is random and does not account for the nodes' locations. However it does not require any priori knowledge among the nodes making this scheme easy to implement.

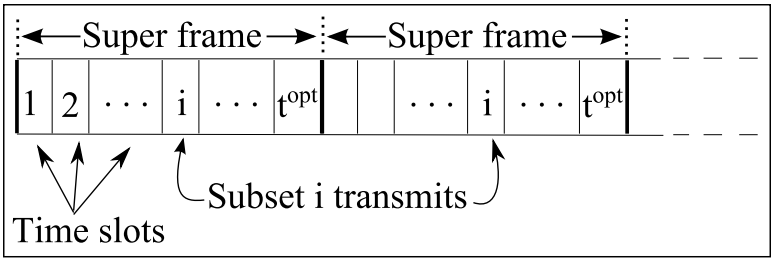


Figure 3.10 TDMA super-frame.

3.3.4 Distributed MAC for Connectivity Maximization of Un-Coordinated DSA Networks

Although knowing t^{opt} is essential for connectivity maximization, however implementation details like frame synchronization, frame and super-frame start and end times, node's time

to transmit, etc, need to be established. MAC protocols provide precise answers to these aspects– a survey of MAC protocols can be found in [76].

Thus, to put our finding about t^{opt} into actual use, we build a MAC protocol around it with the goal of maximizing the connectivity of un-coordinated secondary DSA networks. We acknowledge that, our MAC is different from traditional MAC protocols in i) coordination, where traditional MAC protocols require some level (local or global) of coordination between the nodes regarding neighbor-information exchange, ii) goal, where traditional MAC protocols aim towards increasing the throughput by increasing the number of successful transmissions i.e., increase the pair-wise connectivity. However, this does not increase the network connectivity as the goal in the former is to create independent sets while the latter aims to create one giant component (set) of connected nodes in the network.

3.3.4.1 U-MAC: Un-coordinated MAC

In our MAC, the super-frame is divided into t^{opt} time slots. The beginning of each time slot is marked by a periodic synchronization beacon. We assume that the secondary nodes are synchronized by these beacons. We also include in each super-frame a config-message. This is a bursty-message that is transmitted once at the beginning of each super-frame just before the first beacon. The config-message consists of an update-tone (signal) that is repeated as many as t^{opt} times. The update-tones are placed in a contiguous fashion next to each other as they form the config-message. An update-tone utilizes the same carrier-frequency

of the beacon signal, however it uses an inverted version of the beacon signal i.e., the auto-correlation of the beacon signal and the update-tone is -1. This facilitates the identification of both signals by the secondary node. An illustration of the config-message, update-tone, synchronization beacons, and the corresponding time slots are shown for two super-frames where $t^{opt} = 3$ for the first and $t^{opt} = 2$ for the second in Fig. 3.11.

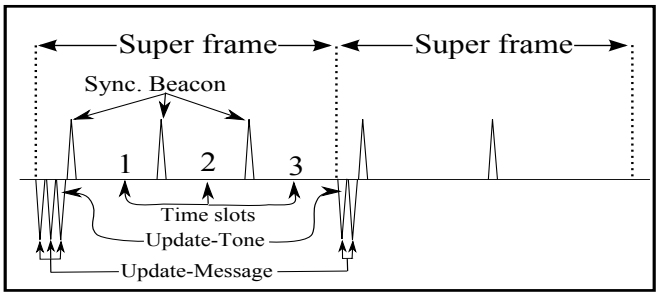


Figure 3.11 Update-tone with $t^{opt} = 3$ followed by 3 synchronization beacons in the first super-frame and $t^{opt} = 2$ followed by 2 synchronization beacons in the second super-frame.

The config-message (via the count of the update-tones) informs the secondary nodes about the number of time slots in each super-frame. Thus, it provides a configuration mechanism such that if any of the network parameters[1] that affects λ_s^{opt} change, then t^{opt} is made known to all nodes. The new value of t^{opt} can be calculated using Eqns. (3.20) and (3.22). The nodes then are informed about the new setting via the config-message in the next super-frame. Overall, the nodes go through 3 phases: sensing, contention and data transmission.

Sensing Phase: In this phase, each node perfectly senses for the following:

[1]These include but not limited to arrival-departure rates, the primary and secondary networks' densities.

1. The config-message which enables the nodes to select its time slot.

2. The beacon which marks the beginning of the node's time slot.

3. The primary user; sensing for the primary user starts after the node hears the beacon that marks its time slot.

Contention/Allocation Phase: Upon hearing the config-message and knowing t^{opt}, the secondary users randomly choose a number between 1 and t^{opt}. This number is the time slot on which the node will transmit its data, i.e., if the chosen time slot is i, then the node will sense for primaries after hearing the i-th beacon. Nodes which have randomly chosen to transmit on the same time-slot will form their subset and their connected component.

Data Transmission Phase: A node which decides that the channel is empty during its time slot, will i) transmit its data to its neighbors, and ii) relay the data of its neighbors. This continues for the entire duration of the time slot.

3.3.4.2 U-MAC Analysis

We analyze the performance of our TDMA-based maximization scheme in terms of some of the commonly used metrics. First, we provide their definitions in our context.

1. Blocking Probability (P_{block}): grabbing a channel does not guarantee a successful communication as the node might be interfered ($SINR < \beta$) by transmissions from other

secondary nodes leading to a unsuccessful communication. With λ_s and λ_p known, we get the blocking probability as:

$$P_{block} = \sum_{M=1}^{4\lambda_p} \int_{R_1=0}^{R_1=r_I+r_p} \int_{\theta_1=0}^{2\pi} \cdots \int_{R_M=0}^{R_1=r_I+r_p} \int_{\theta_M=0}^{2\pi} \sum_{K=0}^{4\lambda_s/t^{opt}} \times \int_{r_1=0}^{r_I} \int_{r_2=0}^{r_I} \cdots \int_{r_K=0}^{r_I} \mathrm{F(x, r_1, \cdot\cdot, r_K, R_1, \cdot\cdot, R_M, \theta_1, \cdot\cdot, \theta_M)} \times \prod_{z'=1}^{M} \frac{2d_{z',i}}{(r_I+r_p)^2} \times \frac{(\mathrm{A_p}\lambda_\mathrm{p})^\mathrm{M}\ \mathrm{e}^{-A\lambda_p}}{M!} \times \prod_{z=1}^{K} \frac{2d_{z,i}}{r_I^2} \times \frac{(\mathrm{A}\lambda_\mathrm{s})^\mathrm{K}\ \mathrm{e}^{-A\lambda_s/t^{opt}}}{K!} \mathrm{d}r_1\, \mathrm{d}r_2\, \cdots\, \mathrm{d}r_K \mathrm{d}R_1\, \mathrm{d}R_2\, \cdots\, \mathrm{d}r_M \mathrm{d}\theta_1\, \mathrm{d}\theta_2\, \cdots\, \mathrm{d}\theta_M \quad (3.23)$$

where A (A_p) is πr_I^2 ($\pi(r_I+r_p)^2$), r_j is a shortcut for $d_{j,i}$. The indicator function $F(\cdot)$ is given by:

$$\mathrm{F}(\cdot) = \begin{cases} 1 & \text{if } \sum_{j=1}^{K} G_j = 0 \text{ or } K = 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.24)$$

where $G_j(\cdot)$ is also an indicator function and is given by:

$$G_j(\cdot) = \begin{cases} 1 \text{ if } \dfrac{\frac{P_s}{r_j^\alpha}}{\sum\limits_{\substack{\forall j' \text{ S.T.} \\ r_{j'} \le r_I, j' \ne j}} \dfrac{\gamma P_s}{r_{j'}^\alpha} + \sum\limits_{\substack{\forall j' \text{ S.T.} \\ d_{z,i} \le r_I}} \dfrac{P_p}{d_{zi}^\alpha}} \ge \beta \\ 0 \text{ otherwise} \end{cases} \quad (3.25)$$

Note, theoretically, for each λ_s the upper bound on the number of secondary users is ∞ as they are Poisson distributed. For better tractability, we bound them with a

maximum value of $4\lambda_s$ which ensures that at least 99.33% of the cases are accounted for as $\sum_{k=0}^{4\lambda_s} \lambda_s^k e^{-\lambda_s}/k! \geq 0.9933$ for $\lambda_s \geq 5$. Eqn. (3.23) uses the law of total probability to iterate through all the possible combinations of distances for node i and its k neighbors. It tests when node i is interfered by its neighbors as well as when it does not have any neighbor in range. This is why summation is from $K = 0$ to $4\lambda_s$. The individual integrations account for the locations of the neighbors one by one. Since Eqn. (3.23) does not have a closed form solution, we resort to numerical analysis for its evaluation.

2. Network Throughput ($\mathcal{R}$): measures the efficiency of the system and is defined as the ratio of the number of packets delivered successfully to the total number of transmitted packets in unit time (time-slot). Thus, for a network with density λ_s, the throughput becomes:

$$\mathcal{R} = P_{grab} \times \left(1 - P_{block}(\frac{\lambda_s}{t^{opt}})\right) \tag{3.26}$$

P_{grab} is the probability of the channel being grabbed by the secondary user i.e., the channel is not utilized by a primary user hence, $P_{grab} = e^{r_I^2 \pi \lambda_p}$.

3.4 Flooding in DSA Networks under the SINR model

In multi-hop wireless network, with no fixed infrastructure the *node outreach* which covers locating resources, advertisement of services, dissemination of information as well as route

and node discovery can be quite challenging. A common solution is flooding as it requires no topological knowledge of the network [33]. The two most common forms of flooding are conventional and probabilistic flooding.

3.4.1 Conventional Flooding

Apart from its simplicity, the main advantage of conventional flooding lies in its robustness i.e., the implicit redundancy built in the algorithm provides resistance against high degree of message losses and node failures [27]. The large number of re-transmitted packets is the major drawback of this technique as each node will retransmit the received packet regardless whether it will aid in the node outreach or not.

In order to reduce the number of re-transmissions, variations of conventional flooding have been proposed [28] like, Time to Live in AODV [29], random walks [30], probabilistic flooding [31], and teeming [32].

3.4.2 Probabilistic Flooding

Probabilistic flooding aims to reduce the total number of rebroadcasts in the network by having a fraction P of each node's neighbors rebroadcast while maintaining enough number of participating nodes for the phase transition to occur. Haas et. al. [31] were one of the earliest to notice that flooding of a gossip in a connected graph under the Boolean model

exhibits bimodal behavior, i.e., there is a critical value of P above which the gossip spreads to almost all the nodes in the network and below which the gossip just dies out. On deeper examination, it turns out randomly choosing a neighbor with probability P corresponds to Poisson thinning [77]. Thus, the network percolates as long as $P\lambda_s > \lambda_c$. The optimization for P were solved via simulation in [31]. A non uniform value of P that is inversely proportional to the node degree is proposed in [27].

In general, flooding techniques rely heavily on network connectivity i.e., as the percentage of connected nodes increases, the broadcasted message propagates further into the network and has a higher chance of being delivered to the destination node. As mentioned earlier, the network's connectivity depends on the network's connection model, as it determines the rule for when/where an edge is established between neighboring nodes.

All the previous work on conventional and probabilistic flooding considered random wireless networks under the Boolean model. However, due to interference tolerance of the primary users, not all the nodes will be able to rebroadcast. This creates inhomogeneity in the spatial distribution of a nodes neighbors, making some of its neighbor(s) more relevant than the others in terms of achieving node outreach which can not be solved by traditional probabilistic flooding.

As we mentioned earlier, in interference-limited networks, the Boolean model is not applicable. However, conventional and probabilistic flooding were never investigated under the SINR model and their performance was left as unanswered question.

3.4.3 Neighbor Aware Probabilistic Flooding: NAPF

In this section, we propose Neighbor Aware Probabilistic Flooding (NAPF) which is our modified version of probabilistic flooding. In doing so, we start by analyzing the mechanisms of probabilistic flooding and its consequences on the node outreach first under the Boolean model and then under the SINR model. Results from both will be used along with concepts from percolation theory to build NAPF.

3.4.3.1 Analyzing Probabilistic Flooding

In a typical probabilistic flooding, each node independently decides whether to forward the broadcast with probability P or not to with probability $1-P$. This gives rise to the following two issues:

1. Neighbors within Close Proximity: A node might have two or more neighbors in close proximity; transmissions by *all* of those neighbors are redundant as just one of them could have covered the same area. In other words, rebroadcasts by physically close neighbors do not necessarily increase the covered area substantially.

   To illustrate, neighbors A and B of node S (all shown in Fig. 3.12-A) are in close proximity of each other. If both A and B decide to rebroadcast the same message from S, one of the rebroadcasts is redundant. The situation will be aggravated even more as the number of neighbors in proximity increases, as the number of redundant

rebroadcasts increase as well. On average, there will be $P\lambda_s\pi r_s^2$ rebroadcasts by the node's neighbors in response to its transmission. Although the redundant rebroadcasts might seem harmless at a first glance but they have the following consequences under both the Boolean and SINR models.

(a) *Boolean Model*: Here, interference free communication can be obtained by the use of orthogonal codes which cost more bandwidth as the number of users increases; another way (to realize the Boolean model) would be through the implementation of a distributed MAC scheme which might suffer from increased waiting time, increased complexity, and increased collisions as the number of users/simultaneous transmissions in the system increases.

(b) *SINR Model*: Contrary to the popular belief that in probabilistic flooding the node reach probability increases with the number of rebroadcasts (caused by increase of P or increase of λ_s or both) [31, 33], we show a different result in Lemma 3.2.

Lemma 3.2. *Under the SINR model, there exists a node density and a rebroadcast probability P which when exceeded results in decreased node outreach.*

Proof: The proof is done by contradiction. Assume that, the node outreach increases with the number of rebroadcasts is true. To keep the proof generic, we assume that there are no primary users. This results in:

$$SINR_{j,i} = \frac{\frac{P_s}{d_{i,j}^{\alpha}}}{\sum_{j'=1}^{P\lambda_s\pi r_s^2-1} \frac{\gamma P_s}{d_{j',i}^{\alpha}} + N_0} \geq \beta \tag{3.27}$$

where N_0 is the Gaussian noise.

Keep in mind, success in flooding under the SINR model, relies on having the SINR $\geq \beta$ at the receiving node, so that the receiving node can rebroadcast it to its neighbors and so on. As λ_s increases, so will the number of rebroadcasts around each relying node. Due to propagation delay and asynchronous transmission by the nodes, the transmissions of those nodes will act as interference to each other's receiver nodes. For a receiving node, increase in λ_s will be reflected by the amount of interference it receives, i.e., the summation term in the denominator of Eqn. (3.27). For some λ_s, with high probability [59], the number of rebroadcasting nodes will be enough to make:

$$\sum_{j'=1}^{P\lambda_s \pi r_s^2 - 1} \frac{\gamma P_s}{d_{j',i}^{\alpha}} + N_0 > P_s / d_{i,j}^{\alpha}$$

which means, the rebroadcasting neighbors will interfere with the receiving nodes. At that point, the receiving nodes will not be able to decode the received signal successfully and hence the message will not be relayed/rebroadcasted any further, thus terminating the flooding process. Under high densities the signal will not even spread to the 2-hop neighbors due to the high interference generated by the transmissions (rebroadcasts) of the first-hop neighbors. This results in a node outreach of zero. Thus a contradiction to the Boolean model. Increasing P (rebroadcasting probability) in probabilistic flooding will increase the total number of rebroadcasting nodes, thus increasing the interference which will also

lead to the same result of zero outreach. □

Note that, $P = 1$ corresponds to conventional flooding.

2. Number of Neighbors and Their Spatial Distribution: When a node rebroadcasts, the ideal case would be to have its neighbors relay that signal in all directions. However due to nodes' spatial distribution as well as the primaries' activity (which prevents any secondary within the coverage radius of the primary from rebroadcasting), the neighbors of a node may be unequally spread around it as shown in Fig. 3.12-B. With probabilistic flooding, a pivotal node (such as node F) might decide not to relay the signal, thus terminating the flooding to the upper part of the network (i.e, node G and others that connect through G). Furthermore, if nodes B and D decide to relay, then there will be unnecessary duplicate broadcasts to the lower part of the network (i.e, to nodes A and C).

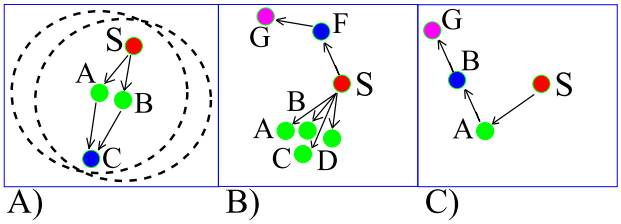


Figure 3.12 A) Nodes A and B in close proximity. B) Inhomogeneous spatial distribution of neighbors. C) Flooding via chain of single neighbors.

3.4.3.2 Number of Neighbors and Percolation Theory

A key observation from percolation theory that contributed to the design of NAPF is that, in order for percolation to occur a node needs a minimum of $\lambda_c \pi r_s^2$ neighbors for the message to flood the network with high probability. At a first glance, this observation urges us to use:

$$P = \begin{cases} \lambda_c \pi r_s^2 / Deg(n_i) & Deg(n_i) > \lambda_c \pi r_s^2 \\ 1 & \text{otherwise} \end{cases} \tag{3.28}$$

where $Deg(n_i)$ is the degree of node n_i. Eqn. (3.28) checks the node's degree and tries to keep the number of rebroadcasting neighbors just enough for percolation to occur. This is done by letting all the node's neighbors to rebroadcast if the node has less than $\lambda_c \pi r_s^2$ neighbors or to have only a fraction of them rebroadcast otherwise. Although such an approach will solve the issue of having a node with a single neighbor (i.e., node S in Fig. 3.12-C), it will not solve the proximity and the in-homogeneity problems. Assume that the minimum degree ($\lambda_c \pi r_s^2$) is 3, and node S in Fig. 3.12-B has a degree of 5. Thus with probability $1 - 3/5$, node F will not rebroadcast and the message will not be flooded to the upper part thus facing the same problem again.

3.4.3.3 Clustering Neighbors

Our distributed solution inherits and employs concepts from probabilistic flooding; however it overcomes its deficiencies by accounting for the close proximity and spatial distribution.

We try a different perspective where we focus on the number of groups of neighbors a node has rather than focusing on its mere number of neighbors (i.e., node's degree). The core idea is that, each transmitting node (broadcasting/rebroadcasting) will classify/cluster its neighbors in groups (using any clustering scheme). The node with the highest degree in each group will be elected as the clusterhead of that group by the transmitting node. The clusterhead will function on behalf its group. Once the node (n_i) which is going to broadcast or rebroadcast determines how many groups it has, denoted by $g(n_i)$, it computes P as:

$$P = \begin{cases} \lambda_c \pi r_s^2 / g(n_i) & g(n_i) > \lambda_c \pi r_s^2 \\ 1 & g(n_i) \leq \lambda_c \pi r_s^2 \end{cases} \tag{3.29}$$

Next, n_i piggy-backs a field of data along the original message (the message to be re-broadcasted). The piggy-backed field contains the value of P along with IDs of the elected clusterheads. We point that n_i's neighbors need not be aware of the clustering that is being done by node n_i. The only way for them to know about the assignments of the cluster heads is when they receive the rebroadcast and the piggy-backed data. In each group, only the cluster head will rebroadcast with the specified probability P. Before it rebroadcasts, it strips the received piggy-back and computes P for its neighbors as well, based on the number of groups it has. Then it piggy-backs its data and rebroadcast to its neighbors. There is an initialization phase where each node exchanges `hello` messages with its neighbors for the purpose of building its neighbor list. If the received signal from a neighbor is higher than some threshold the node marks that neighbor as *in close proximity neighbor* in its neighbor

list. Then the nodes exchange with their first-hop neighbors their lists of neighbors along with the close proximity relations. This information aids each node to cluster its neighbors into groups and elect the cluster heads as it rebroadcasts to them.

In probabilistic flooding, the value P has to be recalculated and updated for all the secondary nodes as λ_p changes. NAPF is designed such that P is calculated for each node based on its number of groups. NAPF will respond automatically to the change in λ_p by changing the value of P for each node according to its degree. Thus adapting for changes in λ_p does not require a global update.

3.5 Connectivity of Multi-Channel DSA Networks

Up to this point, we have investigated and characterized the connectivity of DSA networks under the SINR model on a single channel. However, there are scenarios in which there is more than one channel. In other words, there can be abundance of channels. These scenarios include but are not limited to IEEE 802.22 networks which have about 100 channels of 6 MHz each [78]. Also, the concept of channel fragmentation [79, 80], makes it possible to fragment a single channel into larger number of smaller channels.

Connectivity of DSA networks in the presence of multiple (abundant) channels has not been investigated neither under the Boolean nor the SINR models. Traditionally, the abundance of channels is handled via rendezvous protocols however, it suppresses the effects of such abundance. This is why we use the most naive rendezvous protocol when studying

the effects of such abundance on the connectivity. Such setup corresponds to studying the connectivity of multi-channel random/un-coordinated DSA networks. The study will be done under the Boolean model first so as to isolate the effects of interference, then the SINR model will be considered where the effects of interference and multi-channel availability are considered jointly.

3.5.1 System Model

We consider a network were each secondary user can simultaneously use at most M different channels due to i) each node being equipped with M transceivers (communication modules) [81], [82] or ii) the maximum number of channels that the node can scan per scanning period is M. The scanning limitation has been pointed out in [83] and [84].

The primary network is also generated by a Poisson point process X' with a deployment density of λ_p and coverage radius r_p. The two networks are embedded in $\mathcal{R}^2$ with a total of N channels (vacant and non-vacant), each of which could be accessed by primary users with equal probability i.e., $1/N$. We assume that each primary user is equipped with one transceiver; thus can use only one channel. We point out that, λ_p with 1 channel per user is similar to $\lambda_p/2$ with 2 channels per user. Due to primary activity and spatial diversity, all secondaries do not necessarily observe the same number of vacant channels. The number of vacant/available channels perceived by secondary user i is n_i, $0 \leq n_i \leq N$. Note,

a secondary user S_i can communicate on with a maximum of M of its n_i available channels when $n_i \geq M$. When $n_i < M$, the node can communicate on maximum of n_i channels.

3.5.1.1 Connectivity Conditions

Though there are sophisticated rendezvous protocols [85, 86], we consider a very naive protocol where nodes randomly select a subset of channels from the set of available channels i.e., there is no node-to-node coordination for channel selection. This is to isolate the effects of channel abundance, since choosing an advanced rendezvous scheme lessens its effects.

Under the SINR model, two nodes S_i and S_j are considered to be connected (two-way communication link exists) iff the following two conditions are satisfied.

1. SINR Condition: $SINR_{j,i} \geq \beta$, where β is the receiver threshold. We consider a symmetric two-way communication link between S_i and S_j, therefore $SINR_{i,j}$ must also be greater than or equal to β. Such bi-directional links eliminate the possibility of having an infinite component that it is traversable in one direction only. A pair which has either or both nodes with $SINR < \beta$ for all their common channel(s) will be referred to as *an interfered pair*. The nodes of such pair are invisible to each other.

2. Common Channel(s) Condition: There must be at least one common channel between S_i and S_j, i.e., $Ch(S_i) \cap Ch(S_j) \neq \emptyset$, where $Ch(S_i)$ is the set of randomly selected channels by node S_i from n_i, $0 \leq |Ch(S_i)| \leq M$.

3.5.1.2 Channel Abundance

Due to sensing at different locations, secondary nodes perceive different number of vacant channels, e.g., n_i by S_i and n_j by S_j. It is the subset of the selected channels that determine whether a common channel is guaranteed or not. Thus, we define $n_{i,j}$ as:

$$n_{i,j} = \begin{cases} 0 & n_i \text{ or } n_j = 0 \\ n_i + n_j - n_i \cap n_j & \text{otherwise} \end{cases} \tag{3.30}$$

For $1 \leq n_{i,j} \leq 2M-1$, $Ch(S_i) \cap Ch(S_j) \neq \emptyset$ with probability 1. For example, with $M = 3$ (i.e., two nodes choosing 3 channels independently) there is bound to be a common channel when $1 \leq N \leq 5$ since it leads to $1 \leq n_{i,j} \leq 5$. Examples of $N = 1$, $N = 3$, $N = 5$, and $N = 9$ are illustrated in Fig. 3.13. However, for $n_{i,j} > 2M-1$ (i.e., $n_{i,j} > 5$ when $M = 3$), there is no guarantee on a common channel. For example, consider the case when $N = 9$ with channels numbered from f_1 through f_9. Suppose S_i chooses f_1, f_2, and f_3 as shown in Fig. 3.13. It might so happen that S_j chooses channels f_6, f_8, and f_9 which results in no common channel between the two nodes. This availability of a large number of channels is referred to as 'channel abundance'. Thus, the condition for channel abundance between two nodes S_i and S_j is $M + M \leq n_{i,j}$.

Under channel abundance, it is to be noted that though there could be a number of available channels over which communication link(s) could be established; it is the agreement on the common channel that would establish the link between two nodes. In fact, there are

$\binom{n_{i,j}-M}{M}$ different ways for nodes S_i ad S_j to choose M channels each without having any common channel, assuming n_i, $n_j \geq M$; else the number of ways is $\binom{n_{i,j}-n_i}{n_j}$ for S_j and $\binom{n_{i,j}-n_j}{n_i}$ for S_i. As a consequence, two nodes might remain invisible even if they satisfy the range condition. We want to emphasize that invisibility is *not* due to lack of channels but due to excess of them.

3.5.1.3 Multi-Layered Graph (MLG)

Two neighboring nodes S_i and S_j with $SINR_{i,j} \geq \beta$ and $SINR_{j,i} \geq \beta$ on more than one common channel can be connected in more than one dimension, where each channel is considered a dimension.

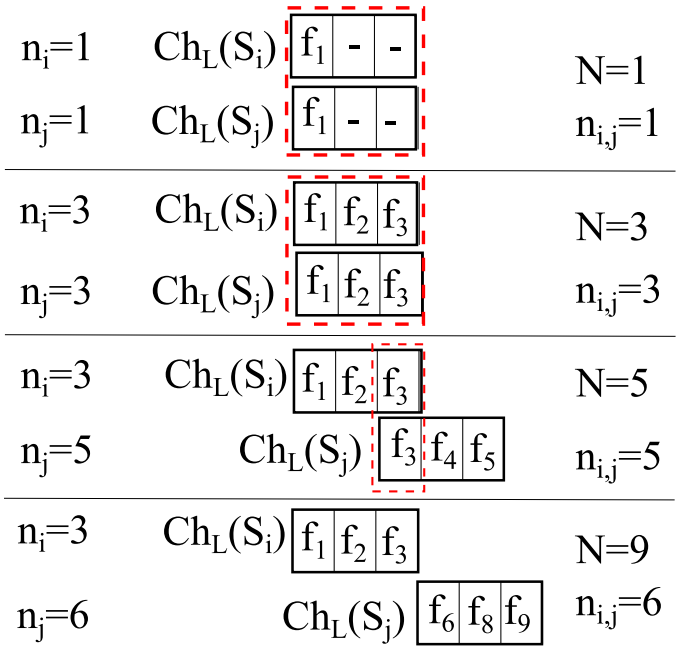


Figure 3.13 Worst case scenarios for two nodes choosing a common channel when $M = 3$.

When $N = 1$, we obtain a traditional graph with a set of secondary nodes and a set of edges (communication links). The edges are established over that single channel/dimension. When $N > 1$, the dimensions can be visualized as independent layers as shown in Figure 3.14. This opens the possibility for the secondary nodes to establish edges in M of the N layers. Clearly, this gives a rise to a Multi-Layered Graph (MLG). Layer i (L_i) of the MLG corresponds to a graph whose edges are established over channel/frequency f_i i.e., $G(V, E_{L_1})$ is the graph with edge connectivity in L_1 (f_1) only.

Since connectivity can be achieved in any of the N layers, we are interested in the Projected-MLG (PMLG) on $\mathcal{R}^2$, which is $G(V, E)$. An edge between two nodes S_i and S_j in the PMLG indicates that: 1) they have a common edge in at least one of the N layers of the MLG and 2) the SINR at both S_i and S_j is greater than β. From a connectivity perspective, it does not matter which layer(s) provide connectivity as long as there is an edge between the two nodes. The reduction of the N-dimensional graph to 1-dimensional PMLG helps us use the concepts of percolation in $\mathcal{R}^2$.

3.5.2 Connectivity Analysis

Now, we proceed to analyze the connectivity and study the effects of changing N, n_{avg} and M and γ. First, we analyze percolation when there is abundance of channels and then we consider percolation with interference.

3.5.2.1 Percolation with Channel Abundance Only ($2M \leq N$, $\gamma = 0$)

Here, we characterize the adverse effects of channel abundance on the network's connectivity. With $2M \leq N$ and $\gamma = 0$, the network's connectivity follows a Boolean model. For two neighboring nodes S_j and S_i, the probability (P_0) of having no common channel between them is:

$$P_0 = \mathrm{P}\Big((\mathrm{M\ channels\ of}\ S_i) \cap (M\ \mathrm{channels\ of}\ S_j) = \emptyset\Big)$$

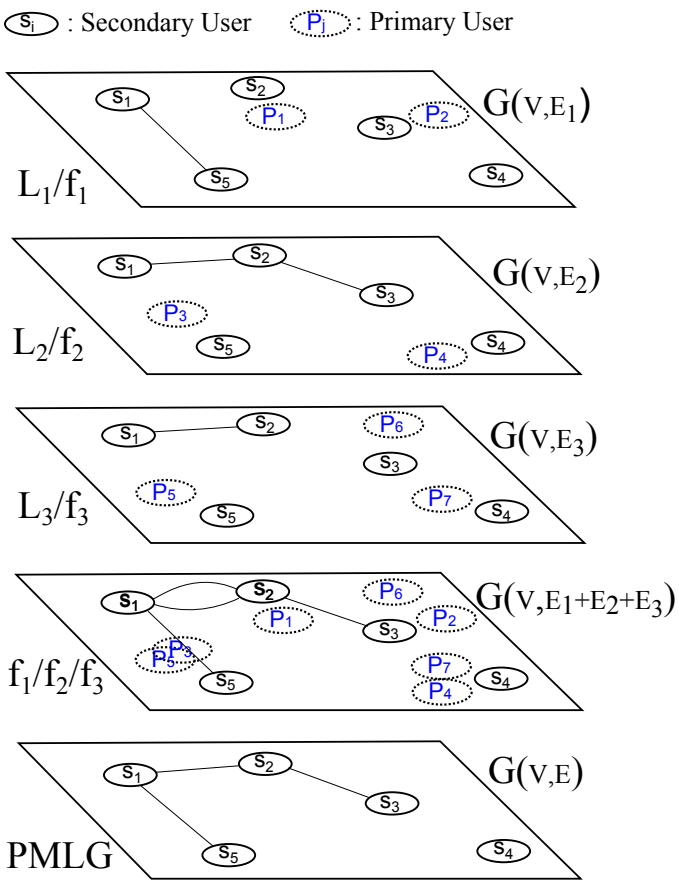


Figure 3.14 MLG with N=3 and its corresponding projection (PMLG)

Further, we partition the range of N into two regions: $[1,\ \ 2M)$ and $[2M,\ \infty]$. For $1 \leq N < 2M$, neighboring nodes are guaranteed to have at least one common channel i.e., $P_0 = 0$. For $N \geq 2M$, a common channel is no longer guaranteed, rather it is probabilistic i.e., $P_0 > 0$. Thus P_0 represents the probability of two neighboring nodes being invisible to each other and can be written as:

$$P_0 = \begin{cases} 0 & 2M > N \\ \frac{\binom{n_{i,j}-M}{M}}{\binom{n_{i,j}}{M}} & 2M \leq N \end{cases} \tag{3.31}$$

Due to this invisibility, the effective density of the Poisson point process is no longer λ_s. Instead it is thinned (Poisson thinning theorem [77]) with probability P_0, resulting in a new density of $\lambda_s \times (1 - P_0)$. This thinning effect is adverse when viewed from a continuum percolation point of view, because percolation occurs when the deployment density exceeds λ_s^c. Thus any reduction in the value of λ_s will adversely affect connectivity and can render a percolated network to a non-percolated one. The thinned secondary network remains percolated as long as:

$$\lambda_s^c < \lambda_s(1 - P_0). \tag{3.32}$$

Note that for a fixed M, as N increases, so does P_0 (more thinning) which reduces $\lambda_s(1 - P_0)$. In Fig. 3.15, we illustrate the thinning effect due to channel abundance on the relative size of the biggest component of a secondary network with $M = 3$ and 250 nodes in an area of 400×400 (i.e., $\lambda_s = 0.0015625$ nodes per unit area). Notably, for $N \in [1, 5]$ the relative size of the biggest component has the same values. For $N > 5$, invisibility occurs

which thins λ_s reducing its connectivity. We define the relative size of the biggest component ($|C|/S$) as the number of nodes in the largest component ($|C|$) to the total number of nodes in the network (S).

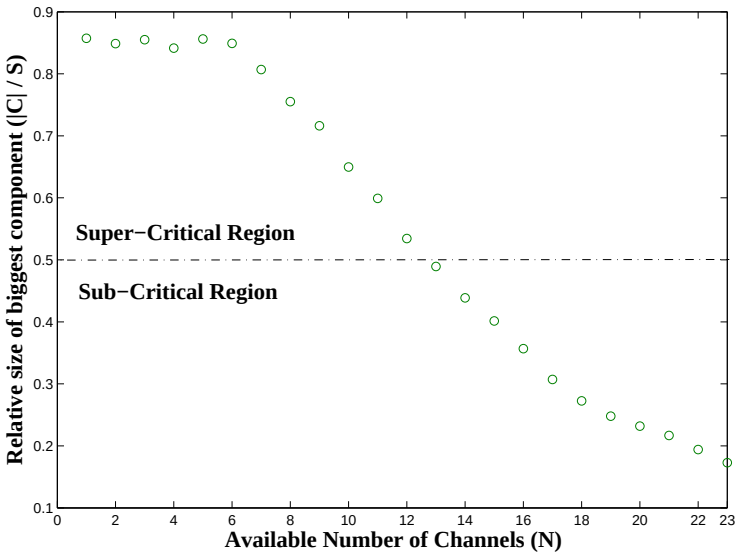


Figure 3.15 Relative size of biggest component for $M = 3$.

3.5.2.2 Percolation with Interference Only ($M = N$, $\gamma \geq 0$)

Here, we characterize the adverse effects of interference only and the rule of γ. With $M = N$ (i.e., no channel abundance) and $\gamma \geq 0$, the network boils down to an equivalent network with $N = M = 1$ because every node is using all the N channels at once.

Suppose K is the average number of interferers a receiver can tolerate before it is interfered. For a single common channel, we define the probability of interference that

violates the SINR condition on that common channel as:

$$P(\text{no. of interferers} > K) = 1 - P(\text{no. of interferes} \leq K)$$

$$= 1 - \sum_{i=0}^{K} \binom{S}{i} \left(\frac{\pi r_I^2}{D_a}\frac{M}{N}\right)^i \left(1 - \left(\frac{\pi r_I^2}{D_a}\frac{M}{N}\right)\right)^{S-i} \tag{3.33}$$

where D_a is the deployment area, and S is the total number of deployed nodes.

For a pair with z common channels, we define their probability of being interfered as:

$$P_{int} = P(\text{Interference on all } z \text{ common channels})$$

$$P_{int} = \left(1 - \sum_{i=0}^{K} \binom{S}{i} \left(\frac{\pi r_I^2}{D_a}\frac{M}{N}\right)^i \left(1 - \left(\frac{\pi r_I^2}{D_a}\frac{M}{N}\right)\right)^{S-i}\right)^z \tag{3.34}$$

Thus interference introduces pair-invisibility which is again modeled as Poisson thinning with probability P_{int}. As a result, the density of the communication-capable nodes becomes $\lambda_s(1 - P_{int})$ instead of λ_s.

In Fig. 3.16 we illustrate the thinning effect due to interference (via controlling γ) on the relative size of the biggest component of a secondary network with $M = N = 1$ and $\lambda_s = 0.0015625$ nodes per unit area. We start with $\gamma = 0$ which is just a pure Boolean model (resulting in maximum connectivity). As $\gamma > 0$, effects of interference start to decrease the SINR values for the pairs, increasing the number of thinned/interfered/invisible pairs. Eventually, the biggest component vanishes.

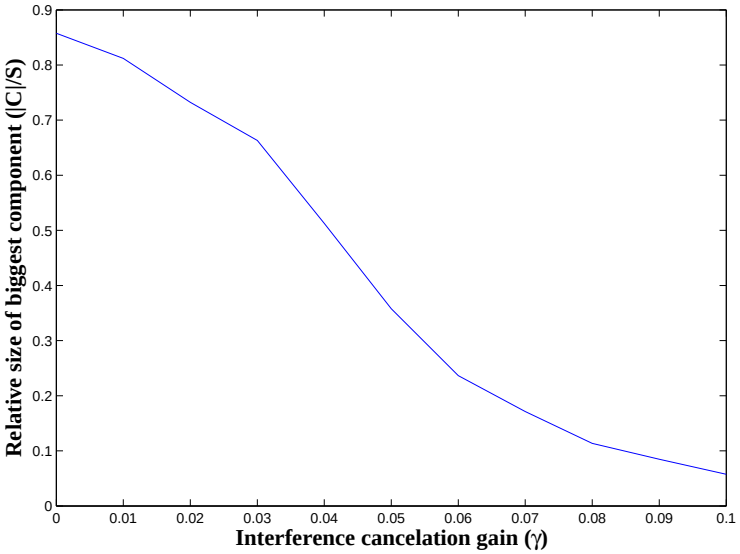


Figure 3.16 Effects of interference on relative size of biggest component.

From Eqn (3.34) it can be noted that for a fixed M, the effect of interference decreases as N increases. The reason is that, the interferers will use diverse channels that decreases the probability of using the same channel being used by the pair. Channel-abundance and interference have opposite reaction to N. For a fixed M, increase in N increases channel-abundance thinning effects (Eqn. 3.31), while interference thinning decreases (Eqn. 3.34).

3.5.3 Percolation in Absence and Presence of Primaries

In this section, we characterize the *combined* effects of channel abundance and interference on the connectivity of the secondary network, First, we consider that there are no primary users i.e., the DSA network functions as a traditional ad hoc network. To do so, we simply set

$\lambda_p = 0$. Then, we consider $\lambda_p > 0$ with zero tolerance for interference; hence the secondary users cannot use any channel that is being used by the primary user(s).

3.5.3.1 Primaries Absent ($\lambda_p = 0$)

When primaries are absent, all the N channels are available to the secondary users resulting in $n_i = n_{i,j} = n_{avg} = N$. Two cases arise: i) $2M > N$ and ii) $2M \leq N$.

1. ***Case 1:*** $2M > N$

   We present two Lemmas that govern the connectivity of the secondary network for $2M > N$.

   Lemma 3.3. *For $2M > N$ a user is bound to have at least one common channel with his in-range neighbor, however the probability of the common channel being interfered by another in range (within r_I) interferer is bounded by* $(0.5,\ 1]$.

   Proof: With $2M > N$, $|Ch(S_i) \cap Ch(S_j)| \geq 1$ for neighboring nodes S_i and S_j which guarantees a common channel. Since an interferer chooses its M channels out of N randomly, its probability (P_{tx}) of transmitting on that common channel is M/N. For $M = N$, $P_{tx} = 1$. As N increases, P_{tx} decreases. In the limiting case, we have:

$$\lim_{M \to +\infty} \frac{M}{2M - 1} = 0.5 \tag{3.35}$$

However for finite values of M, Eqn. (3.35) is strictly greater than 0.5. □

Fig. 3.17 shows the probability of interfering the common channel by an in-range neighbor for $M = 10$ as N is varied from 1 to 19 (i.e., $2M - 1$).

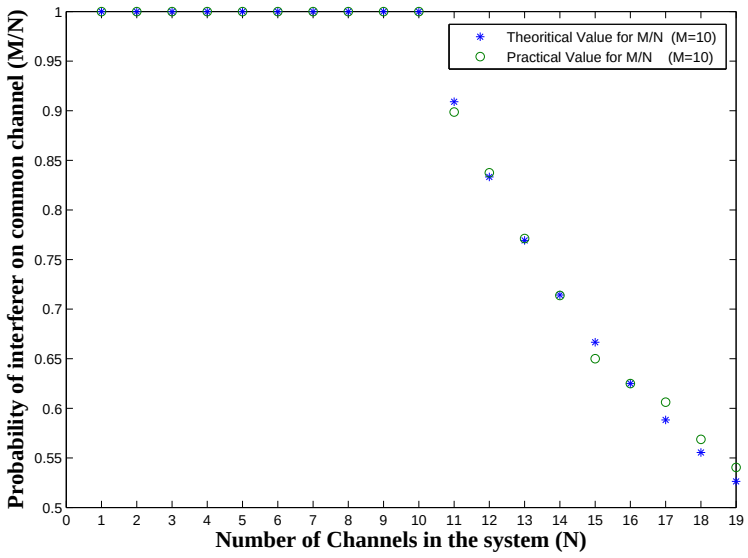


Figure 3.17 Probability of interfering the common channel by an in range neighbor for $M = 4$ and variable N.

Lemma 3.4. *Under the SINR model, the connectivity of a network with density* λ_s, monotonically *increases with the number of channels (N) as long as* $N < 2M$.

Proof: From Lemma 3.3 it was shown that for $N > 2M$, the probability of having an interferer on a common channel decreases as N increases. In other words, fewer nodes will be interfered as N increases resulting in higher connectivity. The increase is monotonic rather than strict because for sparse λ_s, the nodes might not even be in

the vicinity of each other to cause mutual interference. Thus increasing N does not lessen the effects of the interference which results in the same connectivity. □

The above lemma guarantees monotonic increase in connectivity with increase in N as long as $N < 2M$.

2. ***Case 2:*** $2M \leq N$

For $2M \leq N$, a pair will be thinned due to two reasons: interference and channel abundance. Thus a pair can communicate, if it has at least one common channel and the pair is not interfered. Combining these effects, the percolation condition becomes:

$$\lambda_s^c \leq \lambda_s(1 - P_{int})(1 - P_0) \tag{3.36}$$

We use $\mathcal{F}(\cdot)$ to denote the term $(1 - P_{int})(1 - P_0)$ i.e., the combined thinning value. Clearly, $\mathcal{F}(\cdot)$ is a function of N, M, K, r_s, r_I, λ_s and γ, which we write as:

$$\mathcal{F}(\cdot) = \begin{cases} (1 - P_{int})(1 - P_0) & 2M \leq N \\ (1 - P_{int}) & 2M > N \end{cases} \tag{3.37}$$

Proposition 3.1. *Under the SINR model, there exists 3 crucial values for N: N^{opt} (optimal N), N^L (critical lower N) and N^U (critical upper N). N^{opt} yields the maximum connectivity while N^U (N^L) denotes the value of N which when exceeded (not exceeded) leads to the probability of existence of the infinite cluster being 0, i.e.,$\psi_p = 0$ and percolation probability, $\theta_p = 0$.*

To maximize connectivity, thinning has to be minimized. This can be achieved by maximizing the value of $\mathcal{F}(\cdot)$. Suppose $\mathcal{F}(\cdot)$ is maximized for some value of N, say N^{opt}. We can find N^{opt} by solving for the optimization problem:

$$\text{Maximize } (1 - P_{int}) \times (1 - P_0)$$

$$N^{opt} = \arg\max_N \left(1 - \frac{\binom{N\setminus M}{M}}{\binom{N}{M}}\right) \times \left(\sum_{i=0}^{K} \binom{S}{i} \left(\frac{\pi r_I^2}{D_a}\frac{M}{N}\right)^i \times \left(1 - \left(\frac{\pi r_I^2}{D_a}\frac{M}{N}\right)\right)^{S-i}\right) \quad (3.38)$$

N^{opt} represents the boundary between the dominated by interference region and channel abundance region. For $N < N^{opt}$, thinning due to interference will dominate the combined thinning. For $N > N^{opt}$, thinning due to channel-abundance will dominate the combined thinning.

Now, we define N^U (critical N) as $\texttt{inf}(N : \theta_p = 0 \text{ and } N > 1)$ and N^L as $\texttt{sup}(N : \theta_p = 0 \text{ and } N > 1)$. N^L and N^U are found by solving the equation:

$$\lambda_s^c = \lambda_s \times \mathcal{F}(N^c)$$

Solving the above equation we obtain:

$$N^c = \mathcal{F}^{-1}(\lambda_s^c/\lambda_s) \quad (3.39)$$

where $N^L \leq N^{opt} \leq N^U$. Note that the resultant value from Eqn. (3.39) has to be tested against N^{opt} to determine whether it is N^L or N^U. Fig. 3.18 shows theoretical plot for $\mathcal{F}(\cdot)$.

Proposition 3.2. *For a given network with γ, λ_s, r_s and r_I, if the network does not percolate under $N = N^{opt}$, then the network will not percolate for any other value of N.*

Proof: Since N^{opt} results in minimal thinning for λ_s (maximum $\mathcal{F}$), the network will have the largest number of nodes with $SINR \geq \beta$. Then, any other value of N, smaller or larger, results in a smaller value for $\mathcal{F}$ and accordingly less number of nodes. Since the network did not percolate with the maximum number of communication-capable nodes, then it will not percolate under any smaller number of communication-capable nodes. □

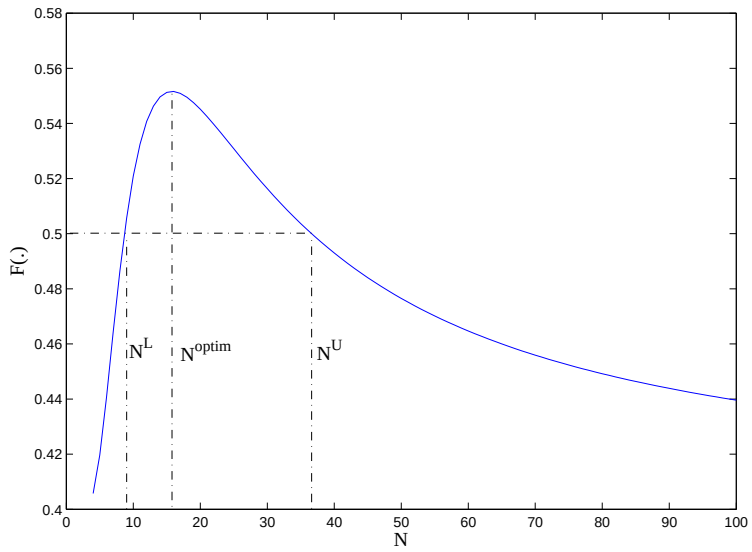


Figure 3.18 $\mathcal{F}(\cdot)$ with N^{opt} and N^c.

3.5.3.2 Primaries Present ($\lambda_p > 0$)

Here, the primary users are present with density λ_p each of whom have radius r_p. We assume each primary user can only use one of the N channels at a time, as a result $n_{avg} \neq N$. Each channel is chosen randomly by the primaries with probability $1/N$. From a channel utilization perspective, this results in thinning the density of X' from λ_p to λ_p/N. In this work, we use zero-interference tolerance form the secondary users (i.e., evict on primary presence).

Contrary to the popular belief that primary users always degrade the connectivity of the secondary users, we present a counter-intuitive observation in Lemma 3.5.

Lemma 3.5. *In the SINR model when $\lambda_p > 0$ and $n_{avg} > N^{opt}$, increase in λ_p increases the connectivity of the secondary network as long as $\lambda_p \leq \lambda_p^{opt}$ and the connectivity of the secondary network is maximized at $\lambda_p = \lambda_p^{opt}$. For any value of n_{avg}, the secondary network remains percolated as long as $\lambda_p^L \leq \lambda_p \leq \lambda_p^U$, where λ_p^U (λ_p^L) is the upper (lower) bound of the primary density, λ_p^{opt} is the optimal primary density.*

λ_p^L, λ^{opt} and λ_p^U are given by:

$$\lambda_p^L = \frac{-N}{A} \log \left(\frac{N^U}{N} \right)$$

$$\lambda_p^{opt} = \frac{-N}{A} \log \left(\frac{N^{opt}}{N} \right)$$

$$\lambda_p^U = \frac{-N}{A} \log \left(\frac{N^L}{N} \right)$$

Of course $\lambda_p^L \leq \lambda_p^{opt} \leq \lambda_p^U$.

Proof: Before the arrival of the first primary user, $n_{avg} = N$. When the first primary user arrives, it randomly selects one of the N available channels making $n_{avg} < N$. As more primary users become active, n_{avg} decreases. This decrease is good as long as the secondary network is in the channel-abundance-thinning region ($N^{opt} < n_{avg}$) since it lowers P_0 (as shown in Eqn. (3.31)). This results in minimizing the thinning of λ_s increasing the connectivity. As λ_p increases, more channels are used by the primaries until $n_{avg} = N^{opt}$ resulting in the maximum connectivity. We denote such value of the primary density with λ_p^{opt} and derive it as follows.

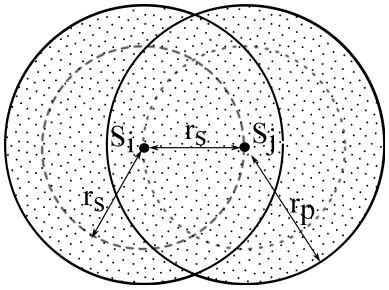


Figure 3.19 Area A in terms of the secondary and primary coverage zones.

Consider two secondary nodes S_i and S_j which are in range of each other and using some channel k as shown in Fig. 3.19. Any primary user using channel k that comes within r_p of S_i or S_j will disrupt communications between S_i and S_j. Thus, the shaded region in Fig. 3.19, referred to as area A, must be void of primary users using channel k.

Let $E_{A,k}^{p>0}$ denote the event that channel k is utilized in area A. The event $E_{A,k}^{p=0}$ is the complement of the event $E_{A,k}^{p>0}$; hence the sum of their probabilities is 1, i.e.,

$$P(E_{A,k}^{p>0}) + P(E_{A,k}^{p=0}) = 1 \tag{3.40}$$

Following an exponential distribution, the probability of no primary arrival, is given by

$$P(E_{A,k}^{p=0}) = \exp\left(\frac{-A\lambda_p}{N}\right) \tag{3.41}$$

From Eqns. (3.40) and (3.41), we get

$$P(E_{A,k}^{p>0}) = 1 - \exp\left(\frac{-A\lambda_p}{N}\right) \tag{3.42}$$

With more primary users, the number of available channels gets thinned (from a secondary user's perspective) with probability P_{thin} which is nothing but the probability of the event $E_{A,k}^{p>0}$. Hence, $P_{thin} = P(E_{A,k}^{p>0})$. To find λ_p^{opt}, we seek the value of λ_p which thins the available number of channels from $n_{avg} = N$ to $n_{avg} = N^{opt}$, thus:

$$(1 - P_{thin}) \times N = N^{opt} \tag{3.43}$$

Substituting for P_{thin} from Eqn. (3.42) results in

$$\exp\left(\frac{-A\lambda_p^{opt}}{N}\right) \times N = N^{opt} \tag{3.44}$$

Solving the above equation for λ_p^{opt} results in:

$$\lambda_p^{opt} = \frac{-N}{A} \log\left(\frac{N^{opt}}{N}\right) \tag{3.45}$$

As we make λ_p greater than λ_p^{opt}, the average number of available channels drops to less than N^{opt}. That pushes the connectivity to the interference-thinning region. Further increase in λ_p reduces n_{avg} even more until $n_{avg} < N^L$ at which percolation disappears. We denote such primary density with λ_p^U. In characterizing λ_p^U, we follow similar analysis for λ_p^{opt}. This results in:

$$\lambda_p^U = \frac{-N}{A} \log\left(\frac{N^L}{N}\right) \tag{3.46}$$

Starting from λ_p^{opt}, if λ_p is gradually decreased, n_{avg} increases with respect to N^{opt} which pushes the connectivity in the channel-abundance-thinning region. Further decrease in λ_p increases n_{avg} even more until $n_{avg} > N^U$ at which percolation disappears. We denote such primary density with λ_p^L which is found to be:

$$\lambda_p^L = \frac{-N}{A} \log\left(\frac{N^U}{N}\right) \tag{3.47}$$

This completes the proof. □

To summarize, the range of λ_p has 2 distinct ranges separated by an optimal point.

1. **Channel Abundance Region** ($0 \leq \lambda_p < \lambda_p^{opt}$)

   This region results in $N^{opt} < n_{avg} \leq \infty$. With such values for n_{avg}, the secondary network is in a channel-abundance dominated thinning mode. Thus, increase in λ_p increases θ_p. This happens because more vacant channels are used by the increasing number of primary users, which in turn decreases n_{avg}. This decreases P_0 thus improving the connectivity θ_p. When λ_p is more than λ_p^L, n_{avg} becomes smaller than N^U at

that primary density, the secondary network percolates. In fact $|C|/S$ keeps increasing until $\lambda_p = \lambda_p^{opt}$.

2. **Optimal Point ($\lambda_p = \lambda_p^{opt}$)**

   There is an optimal number of primary users in terms of eliminating the excess channels resulting in $n_{avg} = N^{opt}$. Thus maximizing the connectivity of the secondary network.

3. **Channel Deprivation Region ($\lambda_p > \lambda_p^{opt}$)**

   This region results in $0 \leq n_{avg} < N^{opt}$. With such values for n_{avg}, increase in λ_p decreases θ_p. This happens because the secondary network is already in the interference-dominant-thinning region ($n_{avg} < N^{opt}$). In other words, with each channel being taken by the primary user, the probability of having the interferer on a pair's common channel increases. This increases the interference on each channel resulting in a thinned pair. Eventually as λ_p increases greater than λ_p^U it results in $n_{avg} < N^L$. At that primary density the network is no longer percolated.

3.5.4 Percolation in Discrete and Continuum spaces

The analyses in the previous section were derived based on concepts of percolation in the SINR model. The MLG is not the typical one-channel SINR model where two nodes are said to be connected if the SINR at both are greater than the threshold. In this section, we show that despite the multi-channels of the MLG, the concepts of percolation still hold

true when applied to the MLG and its projection (PMLG) and it matches those of a typical SINR model.

The domain of the secondary network (represented by the PMLG) is the continuous space $\mathcal{R}^2$. Proving percolation in the continuum model is difficult given the additional constraints of the CRN. However, if we could somehow map the continuum model to a discrete model, we will be able to use the well-known percolation results of the discrete model. Our objective is to couple the continuum percolation model with a two-dimensional discrete lattice $\mathcal{L}$ and show percolation in the lattice implies percolation in $\mathcal{R}^2$. Let us first formally define the discrete lattice and then show the coupling between the two.

- Discrete Grid ($\mathcal{L}$): With $\gamma = 0$, we construct a discrete square lattice $\mathcal{L}$, with distance between neighboring vertices $d > 0$. The center of each edge $q \in \mathcal{L}$ will be denoted x_q, y_q. Let E_q be the event that the edge q is open. Thus E_q occurs iff:

    1. The rectangle $[x_q - 3d/4, x_q + 3d/4] \times [y_q - d/4, y_q + d/4]$ is crossed from left to right by a sequence of neighboring secondary users as shown in Figure 3.20.

    2. The squares $[x_q - 3d/4, x_q - d/4] \times [y_q - d/4, y_q + d/4]$ and $[x_q + d/4, x_q + 3d/4] \times [y_q - d/4, y_q + d/4]$ are crossed from top to bottom by a sequence of neighboring secondary users as shown in Figure 3.20.

    3. For each pair of consecutive/neighboring secondary nodes in the component mentioned above, the range and channel conditions are met i.e., a communication link

is established. The range condition simply means that the nodes are within r_s of each other.

Note that the open vertical edges of $\mathcal{L}$ can be defined by simply rotating the rectangles by 90 degrees. We shift $\mathcal{L}$ by $(d/2, d/2)$ resulting in the dual lattice $\mathcal{L}'$. At this point, the center of a rectangle in $\mathcal{L}$ lies on the side of another rectangle in $\mathcal{L}'$. Thus bond-percolation in $\mathcal{L}'$ results in infinite open path in $\mathcal{L}$.

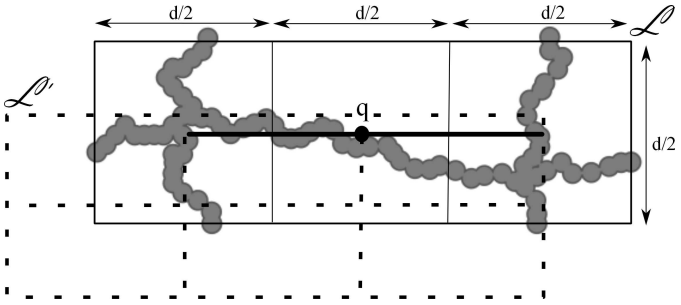


Figure 3.20 Illustration of crossing of the connected component (sequence of neighboring nodes) from left to right and from top to bottom.

- Coupling Percolation of $\mathcal{L}$ and $\mathcal{R}^2$: We use the results from Lemma 1 in [18] which couples percolation in the continuous and discrete models. When percolation occurs in $\mathcal{L}$, an infinite open path appears. The edges in this path correspond to the centers of the rectangles in $\mathcal{L}'$; thus, there exists infinite open path of rectangles. Connectivity of rectangles means satisfaction of conditions 1, 2, and 3 in Section 3.5.4. We emphasize: condition 3 indicates range and channel satisfaction for secondary users that are covered by two adjacent squares along the open edges. For any two adjacent edges, their associated rectangles intersect in the same square of $\mathcal{L}'$. Since both the edges are

open (by assumption of the infiniteness of the open path), there exists a connected component crossing the two rectangles of the edges. The squares of all open edges on $\mathcal{L}'$ are also infinite. Thus, the infinite connected components in $\mathcal{L}$ and $\mathcal{L}'$ imply percolation (infinite path) in the PMLG.

The argument above shows the required conditions for a Boolean model to percolate in multi-dimensions. Relaxing $\gamma = 0$ (i.e., $\gamma > 0$) leads to the SINR model. To show that our multi-dimension SINR model still percolates, we couple the finding about the multi-dimensional Boolean model with the result of [21]. Such coupling shows that percolation can still occur in our model because our argument above showed that, the multi-dimensional Boolean model percolates for some N and M, while [21] shows that, for any λ_s it is always possible to percolate in SINR model with the right selection of γ. Combining both yields percolation in the PMLG. Interested readers can refer to [18] and to [21] for complete proofs.

3.6 Summary

In this chapter, we showed how the traditional percolation condition for continuum networks does not apply to continuum DSA networks under the SINR model. We developed the concept of effective density to account for the percolation visible nodes. We introduced a percolation condition for continuum wireless networks under the SINR model. We elaborated on the concept of the effective density and used it for connectivity maximization via using three different approaches.

Next, we identified the deficiencies of traditional and probabilistic flooding techniques in interference-limited wireless networks. Then, we proposed a modified probabilistic flooding technique that accounts for interference while reduces the duplicate transmissions.

Finally, we studied the connectivity of distributed un-coordinated multi-channel DSA networks under the SINR model. We analyzed the resulting connectivity while accounting for the hardware limitation of the secondary nodes. We showed the resulting connectivity regions and the effects of the primary density on the resulting connectivity of the secondary users.

CHAPTER 4: CAPACITY OF FINITE DSA NETWORKS

In the first part of this chapter, we start by reviewing the capacity maximization problem of DSA networks in the underlay mode and the complexity associated with achieving it. To that end, we derive bounds for that maximum capacity. The in-scalability of the optimization problem motivates us to propose methods which reduce the optimizer's search space. Further, we propose a metric which highlights the importance of considering transmitter-receiver pairs for the optimization. In the second part, we develop a QoS evaluation scheme for the DSA devices. We use vector quantization to identify sub-spaces of the QoS space that yield good QoS. The contents of this chapter appeared in [87, 88, 89]

4.1 Capacity Bounds and Optimizations of Finite DSA Networks

Capacity of a DSA network, in our context, is defined as the sum of data rates that can be achieved by the secondary transmitters. Traditionally capacity maximization of finite DSA networks is solved as a constrained optimization problem[1]. The objective function is:

$$C = \sum_{j=1}^{n/2} \log_2 \left(1 + \frac{a_j P_j}{b_j + \sum_{i \neq j}^{n/2} c_i P_i}\right) \tag{4.1}$$

[1] The constraint is due to the primary interference tolerance.

where a_j is the channel gain between the transmitter-receiver pair j, b_j is the sum of interference from all primary users to receiver j, c_i is the channel gain between the other transmitters and receiver j, and P_x $(1 \leq x \leq n/2)$ are the variables to be optimized. With the constraint being:

$$\sum_{j=1}^{n/2} d_j P_j \leq \gamma \tag{4.2}$$

d_j is the distance between transmitter j and the primary user.

The equation above shows that the optimization is non-convex and is functionally constrained which makes it hard to solve. Moreover, even if the power levels P_j are discretized, the problem is shown to be NP-hard [40, 90] thus scalability and optimization time are two major concerns. Since getting the exact values of the maximum capacity (which involves getting the exact value of P_j's) is a cumbersome process, we revert to bounding the maximum capacity. The bounds provide an insight about the network's maximum and minimum achievable capacities. These can be used as guidelines to predict the network's performance, i.e., if the upper bound of the network's maximum capacity is less than the required *operational* capacity then there is no point of performing any optimization as the network will never meet the requirement.

Moreover, regarding the optimization in Eqn. (4.1) any reduction in the optimizer's search space helps decrease the optimization time and allows for better scalability.

4.1.1 System Model

We consider a DSA network where n secondary transmitters and receivers are scattered randomly over the region of interest in the presence of m primary transmitters. All the primary transmitters tolerate the same level of interference. Half of the secondary nodes (i.e., $n/2$) are transmitters and the other half is receivers. The secondary transmitters are assumed to use the simplest modulation scheme i.e., uncoded binary phase shift keying (BPSK). The secondary receivers employ no noise cancellation technique (i.e., traditional BPSK receiver). The channel between any two nodes is Gaussian with a certain path loss exponent. We consider only one channel with a bandwidth of B Hz. The channel is owned by the primary users and the secondary transmitters access the channel in the underlay mode. Our objective is to bound the maximum number of bits/sec that can be transmitted by the secondary network (i.e., network's capacity) in the underlay mode.

We use $\mathcal{P}_k$, $1 \leq k \leq m$, to refer to the kth primary transceiver. Its location is denoted by $X_{\mathcal{P}_k}$ and power is denoted by $P_{\mathcal{P}_k}$. $\mathcal{R}_i$ is the ith secondary receiver and $\mathcal{T}_j$ is the jth secondary transmitter, $1 \leq i, j \leq n/2$. The location of $\mathcal{R}_i$ ($\mathcal{T}_j$) is denoted by $X_{\mathcal{R}_i}$ ($X_{\mathcal{T}_j}$). $P_{\mathcal{T}_j}$ is the transmit power of $\mathcal{T}_j$.

The Euclidean distance between the secondary transmitter $\mathcal{T}_j$ and primary $\mathcal{P}_k$ is given by $d_{\mathcal{T}_j,\mathcal{P}_k}$. The corresponding channel gain between them is denoted by $G_{\mathcal{T}_j,\mathcal{P}_k} = 1/d^{\alpha}_{\mathcal{T}_j,\mathcal{P}_k}$, where α is the path loss exponent. Similarly $d_{\mathcal{T}_j,\mathcal{R}_i}$ denotes the distance between secondary

transmitter $\mathcal{T}_j$ and secondary receiver $\mathcal{R}_i$. Likewise, $d_{\mathcal{P}_k,\mathcal{R}_i}$ denotes the distance between the primary transceiver $\mathcal{P}_k$ and secondary receiver $\mathcal{R}_i$.

4.1.1.1 Primary Interference Tolerance (γ)

Each primary transceiver/receiver is allowed to tolerate some interference from the secondary transmitters as long as the combined interference is below a certain threshold γ [91]. That is, the total power from *all* the secondary transmitters that arrives to any primary on a channel of B Hz should not exceed γ. Thus, for primary user $\mathcal{P}_k$ we formalize the threshold condition as:

$$\sum_{j=1}^{n/2} P_{\mathcal{T}_j} \times G_{\mathcal{T}_j,\mathcal{P}_k} \leq \gamma \tag{4.3}$$

This threshold condition dictates the maximum power that a secondary transmitters $\mathcal{T}_j$ can use in terms of its closest primary user. For example, if all the power is allocated to only one secondary transmitter, say $\mathcal{T}_1$, Eqn. (4.3) reduces to:

$$P_{\mathcal{T}_1} \times G_{\mathcal{T}_1,\mathcal{P}^{\mathcal{T}_1}} \leq \gamma \tag{4.4}$$

where $\mathcal{P}^{\mathcal{T}_j}$ denotes the primary transceiver that is the closest to $\mathcal{T}_j$. Similarly, $\mathcal{P}^{\mathcal{R}_i}$ denotes the primary transceiver that is the closest to $\mathcal{R}_i$.

Thus, the maximum transmit power ($P_{\mathcal{T}_1}^{max}$) by secondary transmitter $\mathcal{T}_1$ is given by:

$$P_{\mathcal{T}_1}^{max} = \frac{\gamma}{G_{\mathcal{T}_1,\mathcal{P}^{\mathcal{T}_1}}} = \gamma \times d^{\alpha}_{\mathcal{T}_1,\mathcal{P}^{\mathcal{T}_1}} \tag{4.5}$$

Note, when a secondary transmitter $\mathcal{T}_j$ transmits with $P_{\mathcal{T}_j}^{max}$, it implies that there is no leftover power for the other nodes, thus they are turned off.

4.1.1.2 SINR in the Underlay Mode

In the underlay mode, the same channel is used by the primary and the secondary transmitters. This affects the SINR of any secondary transmitter-receiver ($\mathcal{T}_j$,$\mathcal{R}_i$) pair as follows:

$$SINR_{(\mathcal{T}_j,\mathcal{R}_i)} = \frac{\frac{P_{\mathcal{T}_j}}{d^{\alpha}_{\mathcal{T}_j,\mathcal{R}_i}}}{\sum_{j'=1,j'\neq j}^{n/2} \frac{P_{\mathcal{T}_{j'}}}{d^{\alpha}_{\mathcal{T}_{j'},\mathcal{R}_i}} + N_0 + \sum_{k=1}^{m} \frac{P_{\mathcal{P}_k}}{d^{\alpha}_{\mathcal{P}_k,\mathcal{R}_i}}} \tag{4.6}$$

where N_0 is the additive white Gaussian noise.

Shannon's theorem tells us that the capacity of a transmitter-receiver pair increases with their SINR. Therefore, the maximum capacity for any pair $\mathcal{T}_j$-$\mathcal{R}_i$ is achieved with the maximum SINR, denoted by $SINR^{max}_{(\mathcal{T}_j,\mathcal{R}_i)}$. $SINR^{max}_{(\mathcal{T}_j,\mathcal{R}_i)}$ is achieved when $\mathcal{T}_j$ transmits with $P_{\mathcal{T}_j}^{max}$ i.e., $\mathcal{T}_j$ is allocated the entire power budget. This leads to:

$$SINR^{max}_{(\mathcal{T}_j,\mathcal{R}_i)} = \frac{\frac{P^{max}_{\mathcal{T}_j}}{d^{\alpha}_{\mathcal{T}_j,\mathcal{R}_i}}}{N_0 + \sum_{k=1}^{m} \frac{P_{\mathcal{P}_k}}{d^{\alpha}_{\mathcal{P}_k,\mathcal{R}_i}}} \tag{4.7}$$

Substituting $P_{\mathcal{T}_j}^{max}$ from Eqn. (4.5), we get:

$$SINR_{(\mathcal{T}_j,\mathcal{R}_i)}^{max} = \frac{\frac{\gamma \times d_{\mathcal{T}_j,\mathcal{P}}^{\alpha \mathcal{T}_j}}{d_{\mathcal{T}_j,\mathcal{R}_i}^{\alpha}}}{N_0 + \sum_{k=1}^{m} \frac{P_{\mathcal{P}_k}}{d_{\mathcal{P}_k,\mathcal{R}_i}^{\alpha}}} \tag{4.8}$$

For $\sum_{k=1}^{m} \frac{P_{\mathcal{P}_k}}{d_{\mathcal{P}_k,\mathcal{R}_i}^{\alpha}} \gg N_o$, Eqn. (4.8) becomes:

$$SINR_{(\mathcal{T}_j,\mathcal{R}_i)}^{max} = \frac{\frac{\gamma \times d_{\mathcal{T}_j,\mathcal{P}}^{\alpha \mathcal{T}_j}}{d_{\mathcal{T}_j,\mathcal{R}_i}^{\alpha}}}{\sum_{k=1}^{m} \frac{P_{\mathcal{P}_k}}{d_{\mathcal{P}_k,\mathcal{R}_i}^{\alpha}}} \tag{4.9}$$

4.1.2 Capacity Bounds

With the concept of $SINR_{(\mathcal{T}_j,\mathcal{R}_i)}^{max}$ introduced, one of the first questions that comes to the mind is: how to find the bounds on the maximum capacity of a secondary network? One approach could be, to find $SINR_{(\mathcal{T}_j,\mathcal{R}_i)}^{max}$ for all pairs and assign the entire power budget (within the threshold condition) to the pair $\mathcal{T}_{j'}$-$\mathcal{R}_{i'}$ that has the highest $SINR_{(\mathcal{T}_{j'},\mathcal{R}_{i'})}^{max}$ among all the others; the highest $SINR_{\mathcal{T}_j,\mathcal{R}_i}^{max}$ is denoted by $SINR^{max}$.

Although the approach sounds appealing, unfortunately it is not always true. To fully understand, let us consider three transmitter-receiver pairs and assume $SINR_{(\mathcal{T}_3,\mathcal{R}_3)}^{max}$ $< SINR_{(\mathcal{T}_2,\mathcal{R}_2)}^{max} < SINR_{(\mathcal{T}_1,\mathcal{R}_1)}^{max}$. When all the power is assigned to $\mathcal{T}_1$-$\mathcal{R}_1$, it results in

$P_{\mathcal{T}_1} = \gamma \times d^{\alpha}_{\mathcal{T}_1,\mathcal{P}^{\mathcal{T}_1}} = P^{max}_{\mathcal{T}_1}$ and $SINR_{\mathcal{T}_1,\mathcal{R}_1} = SINR^{max}_{(\mathcal{T}_1,\mathcal{R}_1)}$. The resultant capacity is:

$$\begin{aligned} C_{(\mathcal{T}_1\text{-}\mathcal{R}_1,P^{max}_{\mathcal{T}_1})} &= B\log_2\left(1 + SINR^{max}_{(\mathcal{T}_1,\mathcal{R}_1)}\right) \\ &\approx B\log_2\left(SINR^{max}_{(\mathcal{T}_1,\mathcal{R}_1)}\right) \end{aligned}$$

for $SINR^{max}_{(\mathcal{T}_j,\mathcal{R}_i)} \gg 1$. If the pair $\mathcal{T}_1$-$\mathcal{R}_1$ is assigned half the interference tolerance (i.e., $0.5\times\gamma$), it results in $P_{\mathcal{T}_1} = 0.5\times\gamma\times d^{\alpha}_{\mathcal{T}_1,\mathcal{P}^{\mathcal{T}_1}} = 0.5\times P^{max}_{\mathcal{T}_1}$ which in turn leads to $SINR_{\mathcal{T}_1,\mathcal{R}_1} = 0.5 \times SINR^{max}_{(\mathcal{T}_1,\mathcal{R}_1)}$ which affects the capacity as:

$$\begin{aligned} C_{(\mathcal{T}_1\text{-}\mathcal{R}_1,P^{max}_{\mathcal{T}_1}/2)} &= B\log_2\left(0.5 \times SINR^{max}_{(\mathcal{T}_1,\mathcal{R}_1)}\right) \\ &= B\log_2\left(SINR^{max}_{(\mathcal{T}_1,\mathcal{R}_1)}\right) - B \text{ bit/sec} \end{aligned}$$

This means, when the power of a pair $\mathcal{T}_j$-$\mathcal{R}_i$ is halved, its $SINR^{max}_{(\mathcal{T}_j,\mathcal{R}_i)}$ is also halved which reduces the pair's maximum capacity by B bit/sec. Keep in mind, a penalty of B bit/sec is incurred to save half the interference tolerance ($0.5 \times \gamma$). Now the question is: Is it worth to lose B bit/sec to save $0.5 \times \gamma$?

We argue: if the saved $0.5 \times \gamma$ can be invested into another pair $\mathcal{T}_{j'}$-$\mathcal{R}_{i'}$ such that it produces more than B bits/sec, then distributing the power among two pairs is worth. A question immediately arises which is: why halve the power budget; why not split using some other ratio ϵ?

This can be solved as a maximization problem. If we were to maximize $log_2(\epsilon \times A) + log_2((1-\epsilon) \times B)$ where A and B are any two real numbers, then the maximum occurs at $\epsilon = 0.5$.

4.1.2.1 Upper Bound on Maximum Capacity

In a network with one primary ($\mathcal{P}_1$) and 2 secondary pairs, $\mathcal{T}_1$-$\mathcal{R}_1$ and $\mathcal{T}_2$-$\mathcal{R}_2$, with $SINR^{max} = SINR^{max}_{\mathcal{T}_1,\mathcal{R}_1} \geq SINR^{max}_{\mathcal{T}_2,\mathcal{R}_2}$, if investing $0.5 \times \gamma$ in $\mathcal{T}_2$ yields more than the split cost (B bit/sec as discussed above), then the split maximizes the capacity for the two pairs.

We argue that, the capacity of such a network is upper bounded by twice the capacity of $\mathcal{T}_1$-$\mathcal{R}_1$ i.e., as if $\mathcal{T}_2$-$\mathcal{R}_2$ is being replaced by another pair of $\mathcal{T}_1$-$\mathcal{R}_1$. Practically speaking, this scenario occurs when two identical pairs constitute the network. The question is: where would these two identical pairs lie with respect to each other such that the resultant capacity is maximized? Obviously, placing them diametrically opposite the primary (separated by π) will minimize the received interference from each other, thus achieving highest capacity. This is shown in Figure 4.1(a) where $\mathcal{T}_2$-$\mathcal{R}_2$ is being replaced by a copy of $\mathcal{T}_1$-$\mathcal{R}_1$ as shown in Figure 4.1(b) with the same distance from the primary. Notably, this capacity is an upper bound *for the maximum capacity* of all the networks that is formed by placing the same two pairs at various locations, given that $SINR^{max}$ is maintained i.e., $d_{\mathcal{T}_1,\mathcal{R}_1}, d_{\mathcal{T}_1,\mathcal{P}_1}, d_{\mathcal{P}_1,\mathcal{R}_1}$ are all maintained.

To upper bound the maximum capacity of a network with $n/2$ pairs and m primary users, we divide the pairs in groups according to their nearest primary user, i.e., according

to the closest primary from the pair's transmitter. A group (of secondary pairs) will be denoted by g_k where $1 \leq k \leq m$. Thus the nodes of g_k are the closet to $\mathcal{P}_k$. We denote the number of non-empty groups by $\hat{g}$. For each group (g_k) we arrange its pairs in descending order according to their $SINR^{max}_{\mathcal{T}_j,\mathcal{R}_i}$ yielding the ordered set $\mathcal{H}_k = [H_{k_1}, \cdots, H_{k_{n'}}]$, where n' is the last element of the k-th group. Note that in each ordered set $\mathcal{H}_k$, H_{k_1} holds the pair with $SINR^{max}$ value (in g_k).

Similar to the 2-pair network, in each non-empty group, we check whether the capacity can be increased by splitting $\gamma/\hat{g}$ over more than one pair of that group; if so, how many pairs?

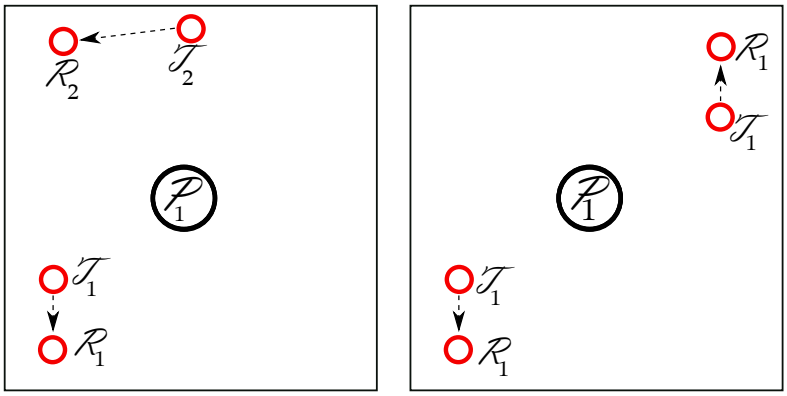


Figure 4.1 A) Network with two pairs; B) $\mathcal{T}_2$-$\mathcal{R}_2$ replaced by a copy of $\mathcal{T}_1$-$\mathcal{R}_1$

Suppose, in each of the g_k groups there are s_k such pairs. Then the s_k pairs of each group will split $\gamma/\hat{g}$. We find s_k in section 4.1.2.1 below. With the number of qualified-for-splitting pairs of each group (s_k) known, we argue that the maximum capacity of the original network is upper bounded by the capacity of a network that contains $\hat{g}$ non-empty groups. In which

each group (g_k) contains s_k copies of its H_{k_1} pair placed with an equi-angular separation of $2\pi/s_k$ around the group's nearest primary user ($\mathcal{P}_k$).

An illustrative example: Let us consider the network shown in Figure 4.2-A. $m = k = 1$ i.e., there is only group g_1 which contains the three pairs. Assume $s_1 = 3$ i.e., (all three pairs are qualified for splitting). Since $\mathcal{T}_1$-$\mathcal{R}_1$ has the $SINR^{max}$ it will be used to replace all the splitting qualified pairs. The resultant network is shown in Fig. 4.2-B. The angular placement ($2\pi/s_1$) makes the distance between the pairs fully characterizable by the extended Pythagorean theorem as well as it makes $SINR_{(\mathcal{T}_j,\mathcal{R}_i)}$ for all the pairs symmetric. Thus the capacity for the network in Fig. 4.2-B is an upper bound for the maximum capacity of the network in Fig. 4.2-A.

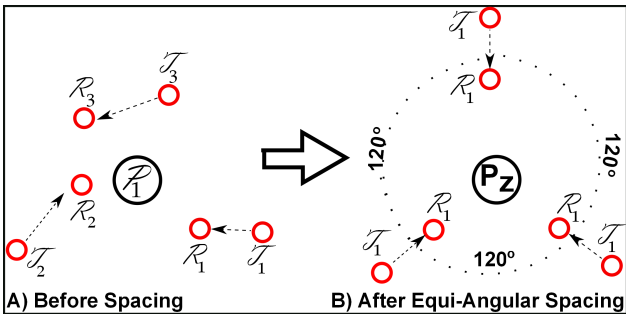


Figure 4.2 A) 3 pairs randomly placed. B) 3 copies of the pair with $SINR^{max}$ are placed with angular separation of 120^o to replace the original 3 pairs.

Number of Qualified pairs per Group (s_k): It has to be noted that in any g_k when s_k pairs qualify for splitting $\gamma/\hat{g}$, they will be replaced by s_k copies of the pair with $SINR^{max}$ of that group (H_{k_1}). Each replaced pair will get $\gamma \times d^{\alpha}_{\mathcal{T}_j,\mathcal{P}^{\mathcal{H}_1}}/(s_k \times \hat{g})$ W of transmission power, thus $SINR_{\mathcal{T}_j,\mathcal{R}_i}$ of each pair is now $SINR^{max}/(s_k \times \hat{g})$ instead of $SINR^{max}$. Thus, in each

g_k there is a trade-off between having more pairs/copies with smaller SINR values or small number of pairs with high SINR values. This trade-off can be stated as an optimization problem:

$$s_k = \arg\max_{A}(A \times log_2(SINR^{max}/(A \times \hat{g}))) \tag{4.10}$$

Solving this equation for each g_k, we find the *upper bound* for its s_k. It is to be noted: even for a symmetrically deployed network as in Figure 4.3-A, it is not necessary that the number of qualified pairs (s_k) is 6 since s_k is determined solely by Eqn. (4.10).

In checking the number of pairs that qualify for the split in each of the groups (s_k), we proceed as follows: in each g_k since the SINR decreases as s_k increases (due to $\gamma/(s_k \times \hat{g})$), each split affects the capacity as $B \times s_k \times \log_2(SINR^{max}/(s_k \times \hat{g})) = B \times s_k \times \log_2 SINR^{max} - B \times s_k \times \log_2(s_k \times \hat{g})$. The term $B \times s \times \log_2(s_k \times \hat{g})$ represents the loss in the capacity of g_k due to incorporating the transmitter of the sth pair of that group.

Using this logic, in each g_k we start with its H_{k_1} ($s_k = 1$) and record the capacity. Next, we check g_k for the pair with the second highest SINR (i.e., H_{k_2}) and split the power over H_{k_1} and H_{k_2}. If splitting over two pairs results in increased bits/sec than the cost of splitting (loss of capacity) then we repeat the process i.e., try to split over with H_{k_3}, H_{k_4}, and so on. The split stopping condition in each group will be either i) the upper bound for s_k is met or, ii) including H_{k_i} results in capacity reduction as compared to the capacity from

using only $i-1$ pairs. Based on this we state: In g_k the s_k-th pair of $\mathcal{H}_k$ qualifies if:

$$B \times \log_2\left(\frac{SINR^{max}_{(\mathcal{T}_s,\mathcal{R}_s)}}{(s_k \times \hat{g})}\right) \geq B \times (s_k - 1)\log_2(s_k \times \hat{g}) \tag{4.11}$$

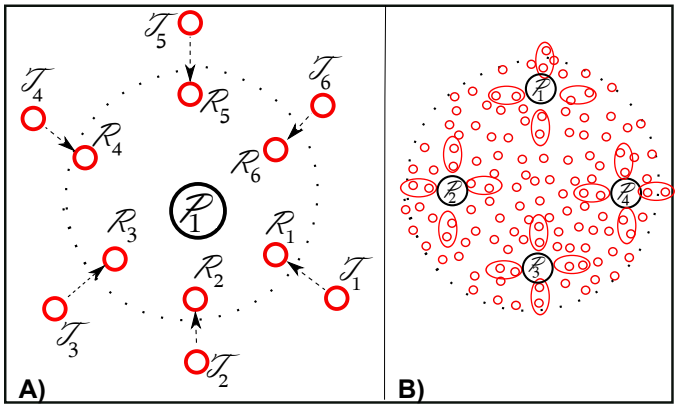


Figure 4.3 A) A network with 6 identical pairs spaced by 60^o. B) A dense network; identical pairs with $SINR^{max}_{(\mathcal{T}_j,\mathcal{R}_i)} = SINR^{max}$ are circled.

4.1.2.2 Putting It All Together

The above discussions can be summarized as follows. Using Eqn. (4.11), the value of s_k is found by iterating over the pairs in $\mathcal{H}_k$ of g_k and checking how many of them qualify for splitting. Each time a pair from the group is qualified, s_k is incremented by 1. Due to the mathematical properties of $s_k \times \log_2(A/s_k)$, the value of s_k has to be upper bounded as was shown in Eqn. (4.10) (A is a constant). Each qualified pair is replaced with a copy of the pair which has the highest $SINR^{max}_{(\mathcal{T}_j,\mathcal{R}_i)}$ in its group i.e., $SINR^{max}$. The copies of each group are placed with equi-angular separation around the group's nearest primary, this

results in identical $SINR^{max}_{(\mathcal{T}_j,\mathcal{R}_i)}$ values for all the group's replaced pairs. The capacity of such a network is an upper bound for the maximum capacity of the original network even when i) the original network is optimized with a global optimizer, and ii) nodes move within their groups given that the pair H_{k_1} still maintains the $SINR^{max}$. The upper bound, C_{UB}, is:

$$C_{UB} = \sum_{k=1}^{\hat{g}} s_k \times B \times \log_2\left(1 + \frac{\frac{SINR^{max}_{H_{k1}}}{\hat{g}\times s_k \times d^{\alpha}_{\mathcal{T}'_k,\mathcal{R}'_k}}}{N'_{k,\mathcal{R}'_k}}\right) \tag{4.12}$$

where $N'_{k,\mathcal{R}'_k}$ is given by:

$$N'_{k,\mathcal{R}'_k} = N_0 + \sum_{k=1}^{m} \frac{P_{\mathcal{P}_k}}{d^{\ \alpha}_{\mathcal{P}_k,\mathcal{R}'_k}} + \sum_{j'=1}^{s_k-1} \frac{SINR^{max}_{H_{k1}}}{\hat{g} \times s_k \times d^{\alpha}_{\mathcal{T}_{\mathcal{R}'_k},\mathcal{R}'_k}}$$

$\mathcal{T}'_k$ ($\mathcal{R}'_k$) are the transmitter (receiver) of the pair H_{k1}. $d_{\mathcal{T}_{j'},\mathcal{R}'_k} = d^2_{\mathcal{P}^{\mathcal{R}'_k},\mathcal{R}'_k} + d^2_{\mathcal{T}_{j'},\mathcal{P}^{\mathcal{T}'_j}} - 2\ d_{\mathcal{P}^{\mathcal{R}'_k},\mathcal{R}'_k} \times d_{\mathcal{T}_{j'},\mathcal{P}^{\mathcal{T}_j}} \times \cos\left(2\pi j^{'}/s_k\right)$.

4.1.2.3 Pairing to Achieve Maximum Capacity

For a given network deployment, the secondary transmitters and receivers can be paired in various ways. Each pairing results in a network topology which when globally optimized yields a maximum capacity that is different from the maximum capacities of the other pairings. This occurs in-spite of having the same number of static nodes. For example, the two networks shown in Figures 4.4(a) and 4.4(b) consist of the same deployment of nodes but with different pairing. Both result in different topologies and different maximum

capacities. The question that arises is: which pairing yields the maximum capacity for a deployed network?

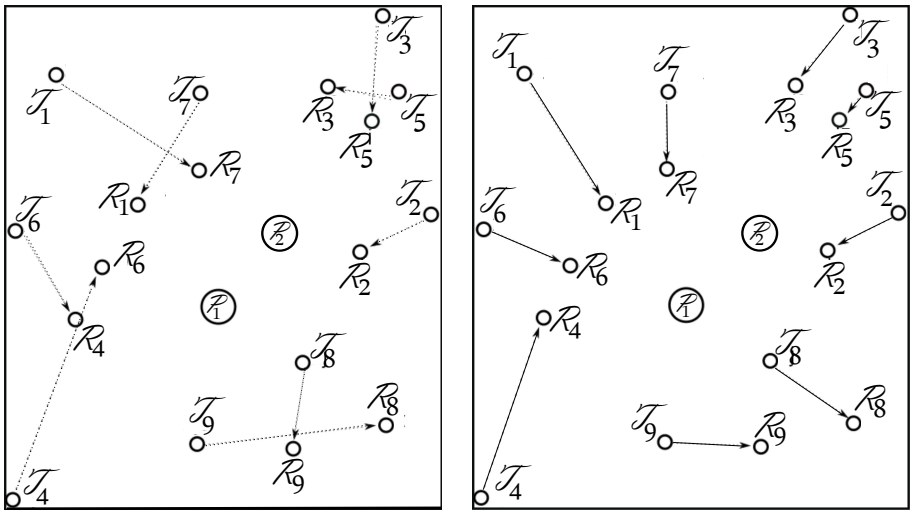


Figure 4.4 Same Network Deployment with two different pairing a) Random pairing. b) Exhaustive search pairing

To answer this question, the pairing relationship itself has to be examined. Suppose the optimizer decides that for the deployment in Figure 4.4(a) $\mathcal{T}_7$-$\mathcal{R}_1$, $\mathcal{T}_2$-$\mathcal{R}_2$ and $\mathcal{T}_6$-$\mathcal{R}_4$ are the only pairs that would yield the maximum capacity for the network; therefore it allocates all the available power-budget to these three pairs. In doing so, $\mathcal{T}_1$ did not get any power allocated, so its receiver $\mathcal{R}_7$ is now idle. Note, compared to $\mathcal{R}_1$, $\mathcal{R}_7$ is closer to $\mathcal{T}_7$, so the total capacity of the network will increase if $\mathcal{T}_7$ transmits to $\mathcal{R}_7$ instead of $\mathcal{R}_1$ ($SINR_{\mathcal{T}_7,\mathcal{R}_7} > SINR_{\mathcal{T}_7,\mathcal{R}_1}$). Now $\mathcal{T}_7$ can pair with $\mathcal{R}_7$, since the latter is idle and transmitting to it increases the system's capacity. To avoid such unstable pairing and to have benchmarks, we propose a pairing scheme that results in the maximum capacity for the *given* deployment.

Tx-Rx Matching Using Exhaustive Search: For a fixed transmit power, the SINR at the receiver of any pair decreases as the transmitter-receiver distance increases. We see a tendency to preserve the transmitted power and hence the SINR, by transmitting to close by neighboring receivers. Such measure will improve the SINR of the pairs. In doing so, the pairs will not revert from each other to achieve a better SINR as each transmitter is already matched to the closest receiver. The question is: how do we implement such pairing (matching) for n nodes?

We explore all possible transmitter-receiver pairings to create various topologies and compute the total capacity for each such topology. The total capacity is nothing but the sum of the individual capacities of every transmitter-receiver pair. We consider the topology for which the system's capacity maximized (i.e., the pairings that constitute that topology).

4.1.2.4 Lower Bound on Maximum Capacity

The lower bound of the maximum capacity represents the guaranteed capacity that results from the deployed DSA network i.e., no matter how in-efficient the power optimizer is, the lower bound is the smallest capacity as it accounts for the worst case scenario. To obtain the lower bound of the maximum capacity, we start with optimizing the objective function:

$$C = B \sum_{j,i=1}^{n/2} \log_2 \left(1 + \frac{\frac{P_{\mathcal{P}_j}}{d^{\alpha}_{\mathcal{T}_j,\mathcal{R}_i}}}{N_0 + \sum_{k=1}^{m} \frac{P_{\mathcal{P}_k}}{d^{\alpha}_{\mathcal{P}_k,\mathcal{R}_i}} + \sum_{j'=1,j'\neq j}^{n/2} \frac{P_{\mathcal{T}_{j'}}}{d^{\alpha}_{\mathcal{T}_{j'},\mathcal{R}_i}}}\right) \tag{4.13}$$

In order to maximize the total capacity, the optimizer does two things. 1) It allocates the entire power budget to one pair at a time and records the $SINR^{max}_{\mathcal{T}_j,\mathcal{R}_i}$ for each pair. The pair with the maximum $SINR^{max}_{\mathcal{T}_j,\mathcal{R}_i}$ (i.e., $SINR^{max}$) yields the highest capacity achievable by a single pair without splitting the power budget. 2) Then it finds whether the capacity can be increased by splitting the power over two or more pairs. It does so by searching for the optimal split ratio (γ/s_k) and the best pairs which together achieve the maximum capacity. Note: the optimizer does not split unless the new split increases the capacity as compared to the current split ratio.

Theoretically, with infinite time, the optimizer will converge to the optimal split and accordingly the best pairs. Practically, with finite time for the optimizer, it might not find the optimal split, resulting in a sub-optimal solution. In terms of optimization-time (convergence time) complexity we address two cases of $\mathcal{O}(1)$ and $\mathcal{O}(n)$ and show the resultant lower bounds for the maximum capacity.

Convergence in $\mathcal{O}(1)$: The smallest time window that can be given to an optimizer is $\mathcal{O}(1)$. In $\mathcal{O}(1)$ no optimization technique will be able to converge to the optimal solution i.e., the winners and the ratio over which γ is to be split. For example, convergence techniques like Nelder-Mead, Lagrange Multiplier, Gradient Methods requires finding the gradient of the objective function (C) and solving its corresponding simultaneous equations which cannot be done in $\mathcal{O}(1)$. Thus, in $\mathcal{O}(1)$, γ has to be allocated to one pair. Moreover that pair has to be chosen randomly. Since allocating γ to any pair $\mathcal{T}_j$-$\mathcal{R}_i$ yields $SINR^{max}_{\mathcal{T}_j,\mathcal{R}_i}$ of that pair, it might so happen that the pair with the worst $SINR^{max}_{\mathcal{T}_j,\mathcal{R}_i}$ value is allocated the entire power

budget which attains the minimum possible maximum capacity given by:

$$C_{LB,\mathcal{O}(1)} = B \times \log_2 (1 + SINR^{lowest}) \tag{4.14}$$

where $SINR^{lowest}$ is the lowest $SINR^{max}_{\mathcal{T}_j,\mathcal{R}_i}$ value among all the $n/2$ pairs. Thus Eqn. (4.14) provides a lower bound on the maximum capacity of the network when optimized in $\mathcal{O}(1)$.

Convergence in $\mathcal{O}(n)$: In this case, there is not enough time to split γ optimally, since splitting involves iterating over the nodes at least twice which requires $\Omega(n^2)$ time. However, $\mathcal{O}(n)$ is enough to iterate all the nodes once. This is enough to locate the pair with the highest SINR value i.e., $SNIR^{max}$. Thus, in terms of $\mathcal{O}(n)$ the lower bound on the maximum capacity of the network, $C_{LB,\mathcal{O}(n)}$, is:

$$C_{LB,\mathcal{O}(n)} = B \times \log_2 (1 + SINR^{max}) \tag{4.15}$$

4.1.3 Elimination Schemes

As we mentioned earlier that, the objective function shown in Eqn. (4.13) is non-convex and even when approximated the problem is still an NP-hard; hence scalability and optimization time are big concerns. This is why, any reduction in the search space helps decrease the optimization time and allows for better scalability.

We note that, the summation in Eqn. (4.13) accounts for the power from *all* the secondary transmitters. As mentioned earlier, this means the power of every $\mathcal{T}_j$ adds an additional dimension/variable to the search space of the optimizer. If we can *somehow* identify the pairs whose powers are always 0, we can exclude them from *the input power vector and eventually from* the optimizer's search space thus reducing the dimensionality. To this end we propose the concept of dead receivers and dead transmitters.

4.1.3.1 Dead Receivers

A receiver $\mathcal{R}_i$ is considered jammed, when its SINR is less than the receiver threshold i.e., when $SINR_{(\mathcal{T}_j,\mathcal{R}_i)} < \beta$. When $\mathcal{R}_i$ is jammed, it will not be able to decode the transmission from its transmitter $\mathcal{T}_j$; such transmissions are considered unsuccessful. Such unsuccessful transmissions have severe consequences– particularly in secondary DSA networks for two important reasons:

1. With a bound the interference the primary is willing to tolerate, any unsuccessful transmission is a waste of power because it leaves less remaining power budget for the other transmitters (recall: there is an interference threshold as given in Eqn. (4.3)).

2. Any unsuccessful transmission adds to the overall noise for the other secondary receivers, thus reducing their SINR and overall capacity. In other words, it is better to put a transmitter to sleep/turn-off if the SINR at its peer-receiver is less than β.

It is intuitive that the power of an unsuccessful transmission could be reassigned to some other successful transmission to increase its SINR. When $SINR_{\mathcal{T}_j,\mathcal{R}_i} < \beta$, it can be tackled by:

i) letting the peer-transmitter $\mathcal{T}_j$ transmits with higher power, given that it is bounded by $P_{\mathcal{T}_j}^{max}$ (Eqn. (4.5)).

ii) reducing the received interference at the receiver $\mathcal{R}_i$ by putting some other secondary transmitters to sleep. It should be noted that, there is no control over the primary transmitter(s), so the interference from the primary user(s) is inevitable.

The two anti-jamming solutions discussed above are conditional in nature, thus it is not possible to determine the exact number of jammed receivers. However, the expected number of jammed receivers can be always found. We are interested in finding those receivers that are always jammed (unconditionally dead). A receiver $\mathcal{R}_i$ is considered a dead-Rx if:

$$SINR_{(\mathcal{T}_j,\mathcal{R}_i)}^{max} < \beta \tag{4.16}$$

The equation above means, receiver $\mathcal{R}_i$ is not able to decode the received signal from $\mathcal{T}_j$ (its transmitter) even if $\mathcal{T}_j$ transmits with $P_{\mathcal{T}_j}^{max}$ because the received interference from the primary transmitter(s) is dominating over $\mathcal{T}_j$'s transmission.

As long as the secondary radio is in the underlay mode (i.e., primary users are ON), there is no point considering the dead receivers and their peer transmitters for any capacity/power optimization. In fact, dropping their decision variable (power request) from the input power

vector reduces the optimizer's search space i.e., instead of solving for an input vector with $n/2$ decision variables (a decision variable is a pair's power), the vector can be reduced to $n/2-$(no. of dead-receivers). Clearly, this reduces the dimensionality of the search space to be explored by the optimizer.

Dead-Rx Zones: Amoeba Like Regions: To find whether receiver $\mathcal{R}_i$ is a dead-Rx or not, its $SINR^{max}_{\mathcal{T}_j,\mathcal{R}_i}$ is compared to β i.e., when its peer-transmitter $\mathcal{T}_j$ transmits with $P^{max}_{\mathcal{T}_j}$, that is:

$$SINR^{max}_{(\mathcal{T}_j,\mathcal{R}_i)} = \frac{\frac{P^{max}_{\mathcal{T}_j}}{d^{\alpha}_{\mathcal{T}_j,\mathcal{R}_i}}}{N_0 + \sum_{k=1}^{m} \frac{P_{\mathcal{P}_k}}{d^{\alpha}_{\mathcal{P}_k,\mathcal{R}_i}}} \geq \beta \tag{4.17}$$

Substituting for $P^{max}_{\mathcal{T}_j}$ yields:

$$\frac{\frac{\gamma \times d^{\alpha}_{\mathcal{T}_j,\mathcal{P}^{\mathcal{T}_j}}}{d^{\alpha}_{\mathcal{T}_j,\mathcal{R}_i}}}{N_0 + \sum_{k=1}^{m} \frac{P_{\mathcal{P}_k}}{d^{\alpha}_{\mathcal{P}_k,\mathcal{R}_i}}} \geq \beta \tag{4.18}$$

Any receiver that does not satisfy the above equation is in the dead-Rx region of one of the primary users. An illustrative example is shown in Figure 4.5-A, where an amoeba like region (the dead-Rx region) contains the dead-Rxs. Such regions exist around the primary users and they can be connected or separated depending on the power, distance, of the primary and secondary users thus they are hard to characterize mathematically. However, the exact number of dead-Rx nodes inside the amoeba region(s) can be found by testing the secondary receivers using Eqn. (4.18). The total number of dead-Rxs in the amoeba

region(s) can be found as:

$$N'_{Rx,Amoeba} = \sum_{i=1}^{n/2} S_i \tag{4.19}$$

$$\text{where,} \quad S_i = \begin{cases} 0 & SINR_{j,i}^{max} \geq \beta \\ 1 & \text{otherwise} \end{cases} \tag{4.20}$$

In Eqn. (4.20), S_i is 0 if the receiver is not dead and 1 if the receiver is dead. Eqn. (4.19) goes through all the $n/2$ pairs, and accumulates the dead ones to obtain $N'_{Rx,Amoeba}$ (number of dead receivers in the amoeba region(s)).

We attempt to characterize the area of the dead-Rx zone(s) using standard geometrical shapes. We approximate using a circle since it is convex and fully characterizable by its radius. Thus for each $\mathcal{P}_k$ we define the *dead-Rx circle* as the biggest circle that can be drawn inside its amoeba region and centered at $X_{\mathcal{P}_k}$. Any secondary receiver that lies inside any of the dead-Rx circle(s) is a dead-Rx. We acknowledge that there are other dead-Rxs which lie outside that dead-Rx circles however they are still inside the amoeba region(s).

Dead-Rx Circle: Having a receiver $\mathcal{R}_i$ inside the dead-Rx circle of $\mathcal{P}^{\mathcal{R}_i}$ (its closest primary transmitter/transceiver) means it remains dead even if all the following conditions are achieved:

1. peer-transmitter $\mathcal{T}_j$ is allocated $P_{\mathcal{T}_j}^{max}$.

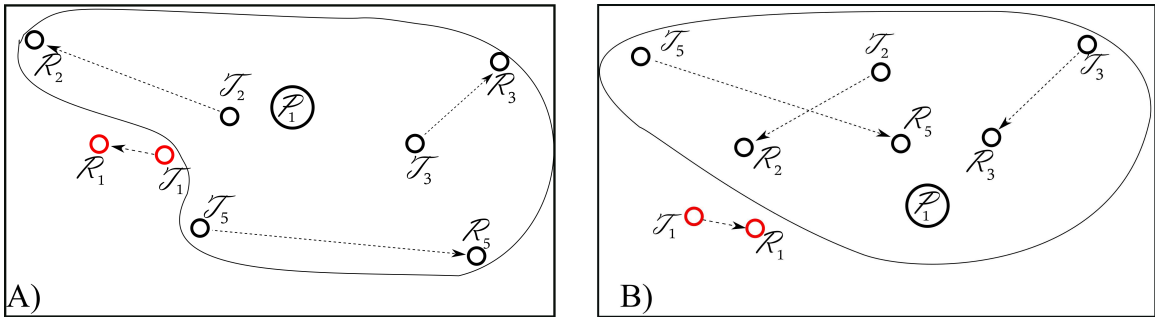


Figure 4.5 A) Visualization of dead-Rx region around the primary receiver/transceiver, dead-Rx nodes lie inside the amoeba; B) Visualization of dead-Tx region, dead-Tx nodes lie inside the amoeba.

2. peer-transmitter $\mathcal{T}_j$ is placed at d_{min} distance from $\mathcal{R}_i$ (which is the closest distance a pair can have). d_{min} denotes the smallest pair-wise distance $d_{\mathcal{T}_j,\mathcal{R}_i}$ among all the pairs and is given by:

$$d_{min} = min(d_{\mathcal{T}_j,\mathcal{R}_i}) \text{ for all the pairs.}$$

3. Only $\mathcal{P}^{\mathcal{R}_i}$ is transmitting and the other primary users are asleep i.e., the received interference from the primary users is at its minimum level.

Note the same logic can be applied for any number of primary users. More primary users will result in more dead nodes as will be shown in the results section.

Dead-Rx Circle: Proof of Concept: The proof is done by induction. When a transmitter $\mathcal{T}_j$ is allocated P_j^{max}, it puts all the other transmitters to sleep according to the threshold condition. Recall, $SINR^{max}_{(\mathcal{T}_j,\mathcal{R}_i)} \propto 1/(d_{\mathcal{T}_j,\mathcal{R}_i})$ and $d_{\mathcal{T}_j,\mathcal{R}_i} \geq d_{min}$ for all the $\mathcal{T}_j$-$\mathcal{R}_i$ pairs. When $d_{\mathcal{T}_j,\mathcal{R}_i}$ of a pair is replaced by d_{min} i.e., the transmitter and receiver are brought closer, it results in a modified maximum SINR ($SINR^{max'}_{(\mathcal{T}_j,\mathcal{R}_i)}$) which is greater than or equal to the

pair's original $SINR^{max}_{(\mathcal{T}_j,\mathcal{R}_i)}$. The allocation of $P_{\mathcal{T}_j} = P^{max}_{\mathcal{T}_j}$ and $d_{\mathcal{T}_j,\mathcal{R}_i} = d_{min}$ gives the secondary pair $\mathcal{T}_j$-$\mathcal{R}_i$ the opportunity to achieve $SINR^{max'}_{(\mathcal{T}_j,\mathcal{R}_i)} > SINR^{max}_{(\mathcal{T}_j,\mathcal{R}_i)}$. If with all these assignments $SINR^{max'}_{(\mathcal{T}_j,\mathcal{R}_i)} < \beta$ i.e., the modified SINR is still less than the secondary receiver threshold, then the receiver $\mathcal{R}_i$ is a dead-Rx. By induction, any other power or pairing relation yields $P_{\mathcal{T}_j} < P^{max}_{\mathcal{T}_j}$ and $d_{\mathcal{T}_j,\mathcal{R}_i} > d_{min}$, which leads to $SINR_{\mathcal{T}_j,\mathcal{R}_i} < SINR^{max'}_{(\mathcal{T}_j,\mathcal{R}_i)}$. Thus, the receiver remains dead because the pairs' current SINR is already smaller than $SINR^{max'}_{(\mathcal{T}_j,\mathcal{R}_i)}$. □

Finding Dead-Rx Radius: For each primary transmitter/transceiver $\mathcal{P}_k$ we characterize its dead-rx circle ($Radius_{Rx_k}$) as follows: a dead-Rx $\mathcal{R}_i$, has $SINR^{max'}_{(\mathcal{T}_j,\mathcal{R}_i)} = \frac{P^{max}_{\mathcal{T}_j} \times G_{\mathcal{T}_j,\mathcal{R}_i}}{P_{\mathcal{P}_k} \times G_{\mathcal{P}_k,\mathcal{R}_i}} < \beta$. Alternatively, $\frac{P_{\mathcal{P}_k} \times G_{\mathcal{P}_k,\mathcal{R}_i}}{P^{max}_{\mathcal{T}_j} \times G_{\mathcal{T}_j,\mathcal{R}_i}} \geq \beta$.

Substituting $P^{max}_{\mathcal{T}_j}$ from Eqn. (4.5), we get

$$\frac{P_{\mathcal{P}_k} \times G_{\mathcal{P}_k,\mathcal{R}_i} \times G_{\mathcal{T}_j,\mathcal{P}_k}}{\gamma \times G_{\mathcal{T}_j,\mathcal{R}_i}} \geq \beta$$

Replacing each channel gain by its corresponding $1/d^{\alpha}$, yields:

$$\frac{P_{\mathcal{P}_k} \times d^{\alpha}_{\mathcal{T}_j,\mathcal{R}_i}}{\beta \times \gamma \times d^{\alpha}_{\mathcal{T}_j,\mathcal{P}_k}} \geq d^{\alpha}_{\mathcal{P}_k,\mathcal{R}_i}$$

To find the dead-Rx radius of $\mathcal{P}_k$, $d_{\mathcal{T}_j,\mathcal{R}_i}$ is set to d_{min} (and all the other primary users are assumed to be a sleep), this leads to:

$$\frac{P_{\mathcal{P}_k} \times {d_{min}}^{\alpha}}{\beta \times \gamma \times d^{\alpha}_{\mathcal{T}_j,\mathcal{P}_k}} \geq d^{\alpha}_{\mathcal{P}_k,\mathcal{R}_i}$$

From the triangle inequality: $d_{\mathcal{T}_j,\mathcal{P}_k} \leq d_{\mathcal{T}_j,\mathcal{R}_i} + d_{\mathcal{P}_k,\mathcal{T}_i}$, we get

$$\frac{P_{\mathcal{P}_k} \times {d_{min}}^{\alpha}}{\beta \times \gamma \times (d_{\mathcal{P}_k,\mathcal{R}_i} + d_{min})^{\alpha}} \geq d^{\alpha}_{\mathcal{P}_k,\mathcal{R}_i}$$

Solving for $d_{\mathcal{P}_k,\mathcal{T}_i}$ results in:

$$d_{\mathcal{P}_k,\mathcal{R}_i} \leq Radius_{Rx_k},\ Radius_{Rx_k} \in R^{+} \tag{4.21}$$

Eqn. (4.21) characterizes a disk whose center is $X_{\mathcal{P}_k}$, and whose radius is $Radius_{Rx_k}$ with $0 \leq k \leq m$. Any secondary receiver which satisfies Eqn. (4.21) is located within the dead-Rx circle of $\mathcal{P}_k$ and is a dead-Rx. The probability of a receiver being in the dead-Rx circle of $\mathcal{P}_k$ is $\frac{\pi \times Radius^2_{Rx_k}}{TotalArea}$. The expected number of dead receivers in the dead-Rx circle of $\mathcal{P}_k$ is:

$$n_{dead-Rx_k} = \frac{n}{2} \times \frac{\pi \times Radius^2_{Rx_k}}{TotalArea} \tag{4.22}$$

4.1.3.2 Dead Transmitters

For each primary receiver/transceiver $\mathcal{P}_k$, we define a dead-Tx zone as the area that covers its closest-in-distance secondary transmitters which cannot establish a successful transmission to their peer-receivers as that would result in violation of primary's threshold condition. Two issues arise:

i) Can a transmitter switch to a lower transmit power if it violates the threshold condition and become a valid transmitter?

ii) A transmitter might not be able to transmit due to other active transmitters that already push the total interference at the primary receiver close to γ, leaving no leftover power to any new transmitter– rendering them dead (from transmission perspective).

An important question is: can we define a dead-Tx taking the above issues into consideration? Though these two issues are probabilistic in nature, we seek those transmitters that are certainly (un-conditionally) not able to transmit in the underlay mode.

We address the first issue by stating: a transmitter $\mathcal{T}_j$ can lower its transmit power $P_{\mathcal{T}_j}$ up to a certain level, which is $P_{\mathcal{T}_j}^{min}$. Since reducing the power reduces the SINR, we have to keep in mind that $SINR_{\mathcal{T}_j,\mathcal{R}_i}$ has to be larger than or equal to β (secondary receiver threshold). However, transmitting with ($P_{\mathcal{T}_j}^{min}$) results in a minimal SINR ($SINR_{\mathcal{T}_j}^{min} = \beta$).

To overcome the probabilistic scope of the second issue, we propose to let the transmitter transmits with $P_{\mathcal{T}_j}^{min}$ alone which lets $\mathcal{T}_j$ causes the minimal interference possible for the closest primary receiver/transceiver $\mathcal{P}^{\mathcal{T}_j}$. If with all these facilitations, the transmitter's

$P^{min}_{\mathcal{T}_j}$ is still violating the threshold condition, i.e., $P^{min}_{\mathcal{T}_j} \times G_{\mathcal{T}_j,\mathcal{P}^{\mathcal{T}_j}} > \gamma$, the transmitter $\mathcal{T}_j$ is declared as a dead-Tx.

Dead-Tx Zones: Amoeba Like Regions: The main reason for obtaining an amoeba like region for a dead-Tx zone, is the difference between $d_{\mathcal{T}_j,\mathcal{P}^{\mathcal{T}_j}}$, $d_{\mathcal{T}_j,\mathcal{R}_i}$ for all the pairs. For example in Figure 4.5-B, although $\mathcal{T}_1$ is closer to the primary than $\mathcal{T}_5$, the distance $d_{\mathcal{T}_5,\mathcal{R}_5} > d_{\mathcal{T}_1,\mathcal{R}_1}$. Therefore, $\mathcal{T}_5$ cannot even sustain $SINR^{min}_{(\mathcal{T}_5,\mathcal{R}_5)} = \beta$, so it is a dead-Tx. The same applies to $\mathcal{T}_2$ and $\mathcal{T}_3$. However, $\mathcal{T}_5$ will not be a dead-Tx if it transmits to $\mathcal{R}_1$. It is clear that the concept of conditionality has to be accounted for when dealing with the dead-Txs– a transmitter might be considered dead if it pairs with a specific set of receiver(s) and not dead otherwise. Such conditionality results in the amoeba like region(s) around the primary receiver(s). To have a successful transmission from $\mathcal{T}_j$, it should have:

$$P^{min}_{\mathcal{T}_j} \leq P_{\mathcal{T}_j} \leq P^{max}_{\mathcal{T}_j}$$

For $P^{min}_{\mathcal{T}_j}$ the definition states: $SINR^{min}_{\mathcal{T}_j,\mathcal{R}_i} = \beta$. Substituting for $SINR^{min}_{\mathcal{T}_j,\mathcal{R}_i}$ yields:

$$\frac{P^{min}_{\mathcal{T}_j} \times G_{\mathcal{T}_j,\mathcal{R}_i}}{\sum_{k=1}^{m} P_{\mathcal{P}_k} \times G_{\mathcal{P}_k,i}} = \beta$$

$$\text{Or,} \qquad P^{min}_{\mathcal{T}_j} = \beta \times d^{\alpha}_{\mathcal{T}_j,\mathcal{R}_i} \times \sum_{k=1}^{m} \frac{P_{\mathcal{P}_k}}{d^{\alpha}_{\mathcal{P}_k,\mathcal{R}_i}} \tag{4.23}$$

Transmitter $\mathcal{T}_j$ is dead-Tx if it satisfies:

$$P_{\mathcal{T}_j}^{min} \times G_{\mathcal{T}_j,\mathcal{P}^{\mathcal{T}_j}} > \gamma \tag{4.24}$$

The exact number of dead-Txs in the Amoeba is:

$$N_{Tx,Amoeba}^{'} = \sum_{j=1}^{n/2} S_j \tag{4.25}$$

$$\text{where,} \quad S_j = \begin{cases} 0 & \gamma > P_{\mathcal{T}_j}^{min} \times G_{\mathcal{T}_j,\mathcal{P}^{\mathcal{T}_j}} \\ 1 & \text{otherwise} \end{cases} \tag{4.26}$$

S_j is 1 if the node is a dead and 0 otherwise.

Dead-Tx Circle: Characterizing the area of the amoeba is not an easy task, so we switch to finding the dead-Tx circle around each primary receiver/transceiver since it is fully characterized by its radius. Having a transmitter $\mathcal{T}_j$ inside any of the dead-Tx circle(s) means it will violate the threshold condition of that primary transceiver even all the following conditions are satisfied:

1. let it transmit with $P_{\mathcal{T}_j}^{min}$ and put all the other transmitters to sleep.

2. place its peer-receiver at a distance of $\mathcal{R}_i$ at d_{min} from it (which is the closest distance any pair can have for a given network deployment).

3. Put all the primary transmitters to sleep except $\mathcal{P}^{\mathcal{T}_j}$. This last condition ensure a minimal primary interference to the pair $\mathcal{T}_j$-$\mathcal{R}_i$.

Note, the same logic applies for any number of primary users (transceivers). As the number of considered primary users increases, so does the number of dead transmitters.

Dead-Tx Circle: Proof of Concept: The proof is done by induction. When $P_{\mathcal{T}_j}^{min}$ is assigned to $\mathcal{T}_j$ and all the other secondary transmitters are put to sleep as well as all the primary transmitters except the closets one ($\mathcal{P}^{\mathcal{T}_j}$). This means that the current transmitter is the only interferer to the primary receiver/transceiver. Since $P_{\mathcal{T}_j}^{min} \propto d_{\mathcal{T}_j,\mathcal{R}_i}$, the peer receiver $\mathcal{R}_i$ should be placed as close as possible to transmitter $\mathcal{T}_j$. The closest pair-wise distance in the network is d_{min}. For that reason the receiver $\mathcal{R}_i$ will be placed d_{min} away from $\mathcal{T}_j$. Doing so, gives the secondary transmitter $\mathcal{T}_j$ the opportunity to transmit with the smallest $P_{\mathcal{T}_j}^{min}$. If with all the mentioned assignments (which aids in minimizing the transmitted power), $\mathcal{T}_j$ is still dead (i.e., $P_{\mathcal{T}_j}^{min} \times G_{\mathcal{T}_j,\mathcal{P}^{\mathcal{T}_j}} > \gamma$), it will remain dead no matter what, because:

1. $\mathcal{T}_j$ will be paired with another receiver $\mathcal{R}_i^{'}$ such that $d_{\mathcal{T}_j,\mathcal{R}_i^{'}} \geq d_{min}$

2. When the other secondary transmitters and primary transmitters are allowed to transmit as well, they add interference to $\mathcal{T}_j$'s receiver *which brings the SINR of the pair even lower.*

It is clear that, either (i) or (ii) makes $P_{\mathcal{T}_j} > P_{\mathcal{T}_j}^{min}$. Thus $\mathcal{T}_j$ will keep on violating the threshold condition for any other power value. □

Finding Dead-Tx Radius: Similar analysis to those used in finding the dead-Rx radius will be used in finding the dead-Tx radius for each $\mathcal{P}_k$.

From Eqn. (4.24), a dead-Tx has $\frac{\frac{\beta \times P_{\mathcal{P}_k} \times d^{\alpha}_{\mathcal{T}_j,\mathcal{R}_i}}{d^{\alpha}_{\mathcal{P}_k,\mathcal{R}_i}}}{d^{\alpha}_{\mathcal{T}_j,\mathcal{P}_k}} > \gamma$. By triangulation inequality, we have $d_{\mathcal{P}_k,\mathcal{R}_i} \leq d_{\mathcal{T}_j,\mathcal{R}_i} + d_{\mathcal{T}_j,\mathcal{R}_k}$. Using the definition of the dead-Tx circle (i.e., $d_{\mathcal{T}_j,\mathcal{R}_i} = d_{min}$), we get:

$$\frac{\frac{\beta \times P_{\mathcal{P}_k} \times {d_{min}}^{\alpha}}{(d_{\mathcal{T}_j,\mathcal{R}_k} + d_{min})^{\alpha}}}{{d\mathcal{T}_j, \mathcal{P}_k}^{\alpha}} > \gamma \tag{4.27}$$

Solving for $d_{\mathcal{T}_j,\mathcal{P}_k}$ results in:

$$d_{\mathcal{T}_j,\mathcal{P}_k} \leq Radius_{Tx_k},\ Radius_{Tx_k} \in R^{+} \tag{4.28}$$

Eqn. (4.28) characterizes a disk centered at $X_{\mathcal{P}_k}$ and whose radius is $Radius_{Tx_k}$. Any secondary transmitter that satisfies Eqn. (4.28) is located inside the dead-Tx circle of $\mathcal{P}_k$ and is a dead-Tx. Accordingly, the expected number of dead transmitters in the dead-Tx circle of $\mathcal{P}_k$ is:

$$n_{dead-Tx_k} = \frac{n}{2} \times \frac{\pi \times Radius^2_{Tx_k}}{TotalArea} \tag{4.29}$$

4.1.4 Pre-processing for Optimizations

A DSA network is unlike a legacy wireless network where transmitter $\mathcal{T}_j^{'}$ and receiver $\mathcal{R}_i^{'}$ can communicate successfully as long as $SINR_{\mathcal{T}_j^{'},\mathcal{R}_i^{'}} \geq \beta$, regardless of the distance between them as well as the level of interference at the receiver. Thus in a legacy network, any transmitter-

receiver pairing can be considered feasible with the right transmit power (we acknowledge the hardware limitation; our treatment is general). As we mentioned earlier, in DSA networks the transmit power $P_{\mathcal{T}_j}$ of transmitter $\mathcal{T}_j$ can be as high as $P_{\mathcal{T}_j}^{max}$ according to Eqn. (4.5), which puts a limit on the coverage area of $\mathcal{T}_j$. With more active secondary transmitters, the coverage area shrinks because: i) other secondary transmitters act as interferes to the pair $\mathcal{T}_j$-$\mathcal{R}_i$, and ii) other active secondary transmitters contribute to the total noise perceived by the primary receiver (transceiver), thus preventing $\mathcal{T}_j$ from transmitting with higher power. This observation allows us to exclude some pairs from the optimizer's search space without compromising the optimality. Let us first discuss the optimization problem then proceed with further dimensionality reduction.

The Optimization Problem: The optimization here is to optimize the transmit power of each pair such that the total network capacity is maximized given a power budget. Thus an optimizer can be considered as a power control algorithm that maximizes the capacity. We treat the power control algorithm as a black-box that receives power allocation requests in the form of an input vector (of length $L \leq n/2$) $[P_1, P_2, ..., P_L]$, where L is the number of active transmitters. At this point, a request $P_{\mathcal{T}_j}$ (by $\mathcal{T}_j$) is a member/dimension of the vector space which spans the optimizer's search space. Larger input vectors (i.e., big values of L) require more time to converge/solve.

Coverage Zones: In searching for additional pair-elimination rules, we note that a paring relation $\mathcal{T}_j$-$\mathcal{R}_i$ is valid if receiver $\mathcal{R}_i$ is inside the feasible region (coverage zone) of its peer-transmitter $\mathcal{T}_j$, otherwise it is in the dead region. We use this observation for further

reducing the optimizer's state (search) space. It can be noted that the coverage zone $Z_{\mathcal{T}_j}$ of $\mathcal{T}_j$ is bounded by $[0, Z^{max}_{\mathcal{T}_j}]$, where $Z^{max}_{\mathcal{T}_j}$ (maximum coverage zone for $\mathcal{T}_j$) is obtained when $\mathcal{T}_j$ is assigned $P^{max}_{\mathcal{T}_j}$. Now, the question is: What value for $Z_{\mathcal{T}_j}$ should be considered in deciding whether $\mathcal{R}_i$ is inside $Z_{\mathcal{T}_j}$ or not? We argue that if receiver $\mathcal{R}_i$ is outside $Z^{max}_{\mathcal{T}_j}$, it is dead, and such a pair $\mathcal{T}_j$-$\mathcal{R}_i$ can safely be excluded from the optimizer's search space. That happens if $SINR^{max}_{\mathcal{T}_j,\mathcal{R}_i} \leq \beta$. That is:

$$\frac{\frac{P^{max}_{\mathcal{T}_j}}{d^{\alpha}_{\mathcal{T}_j,\mathcal{R}_i}}}{N_0 + \sum_{k=1}^{m} \frac{P_{\mathcal{P}_k}}{d^{\alpha}_{\mathcal{P}_k,\mathcal{R}_i}}} = \frac{\frac{\gamma \times d^{\alpha}_{\mathcal{T}_j,\mathcal{P}^{\mathcal{T}_j}}}{d^{\alpha}_{\mathcal{T}_j,\mathcal{R}_i}}}{N_0 + \sum_{k=1}^{m} \frac{P_{\mathcal{P}_k}}{d^{\alpha}_{\mathcal{P}_k,\mathcal{R}_i}}} \leq \beta \tag{4.30}$$

The expression on the right side follows for $N_0 \ll \sum_{k=1}^{m} P_{\mathcal{P}_k}/d^{\alpha}_{\mathcal{P}_k,\mathcal{R}_i}$.

If a receiver $\mathcal{R}_i$ satisfies Eqn. (4.30), it is said to be outside $Z^{max}_{\mathcal{T}_j}$; hence the pair $\mathcal{T}_j$-$\mathcal{R}_i$ can be excluded. We acknowledge that the existence of a secondary receiver $\mathcal{R}_i$ inside $Z^{max}_{\mathcal{T}_j}$ does not necessarily mean $\mathcal{R}_i$ is alive, because $\mathcal{T}_j$ may not be transmitting with $P^{max}_{\mathcal{T}_j}$, thus $Z_{\mathcal{T}_j} < Z^{max}_{\mathcal{T}_j}$, which does not reveal whether $\mathcal{R}_i$ is dead or alive.

To summarize, the elimination rules are:

1. Exclude the power request for any pair whose transmitter is a dead-Tx.

2. Exclude the power request for any pair whose receiver is a dead-Rx.

3. Exclude the power request for any pair which satisfies Eqn. (4.30).

Any pair verifying any of the conditions above will be marked as a *dead-pair*.

Since dead pairs are orthogonal to the output decision vector (i.e., assigned 0 power), there is no point submitting their requests to the power control algorithm (optimizer). On the contrary, not considering them helps in reducing the dimensionality of the search space. This motivates us to purge the input requests vector before it is fed to the optimizer. Instead of submitting and optimizing a vector of $n/2$ requests (dimensions) we propose to submit and optimize a reduced vector of length $n/2 - n_{dead}$, where n_{dead} is the number of dead pairs.

4.1.4.1 Relative Goodness of Tx-Rx Pairs

Global optimization for the capacity of the entire network becomes infeasible when the number of transmitter-receiver pairs is too large for the optimizer to compute within a specific amount of time (user time budget). To that end, we propose a metric $SINR^{Rel}$ which abstracts the relative distance relations between each pair and the primary user(s) as well. $SINR^{Rel}$ provides a measure of the goodness of a pair's ability to contribute towards the system capacity. Thus it reflects the pair's potential for "winning" power allocation from the optimizer.

Defining $SINR^{Rel}_{\mathcal{T}_j,\mathcal{R}_i} = SINR^{max}_{(\mathcal{T}_j,\mathcal{R}_i)} = \frac{P^{max}_{\mathcal{T}_j}/d^{\alpha}_{\mathcal{T}_j,\mathcal{R}_i}}{\sum_{k=1}^{m} P_{\mathcal{P}_k}/d^{\alpha}_{\mathcal{P}_k,\mathcal{R}_i}}$ with $P^{max}_{\mathcal{T}_j} = \gamma \times d^{\alpha}_{\mathcal{T}_j,\mathcal{P}^{\mathcal{T}_j}}$ sounds appealing. This is because the numerator specifies the maximum power that can be received at $\mathcal{R}_i$ from $\mathcal{T}_j$, and the denominator specifies the interference from all the primary transmitter(s). However, that makes the metric more of a conditional measure as it is conditioned on $P_{\mathcal{T}_j} = P^{max}_{\mathcal{T}_j}$, which means it does not account for interference from other live pairs.

At this point, we point out that the interference from the other live transmitters is not known because i) this step precedes the optimization and ii) the optimization itself might not be feasible. As a result, we resort to an alternative method.

A pair $\mathcal{T}_{j'}$-$\mathcal{R}_{i'}$ with high $SINR^{max}_{\mathcal{T}_{j'},\mathcal{R}_{i'}}$ is seen as a good candidate (capacity maximizer) by the optimizer. So when the receiver $\mathcal{R}_i$ (receiver of $\mathcal{T}_j$) is close to a conditional winner $\mathcal{T}_{j'}$, $\mathcal{T}_{j'}$ becomes a candidate interferer to $\mathcal{R}_i$. The closer $\mathcal{R}_i$ gets to $\mathcal{T}_{j'}$, the higher the interference from $\mathcal{T}_{j'}$ to $\mathcal{R}_i$ (in case $\mathcal{T}_{j'}$ gets power from the optimizer). So $SINR^{Rel}_{\mathcal{T}_j,\mathcal{R}_i}$ should decrease due to the existence of $\mathcal{T}_{j'}$. If $\mathcal{R}_i$ is far from such a transmitter-receiver pair, $\mathcal{R}_i$'s metric should increase reflecting the relation of $\mathcal{T}_j$-$\mathcal{R}_i$ with the other good candidates around them. Continuing with the same approach, we define:

$$SINR^{Rel}_{\mathcal{T}_j,\mathcal{R}_i} = \frac{P^{max}_{\mathcal{T}_j}/d^{\alpha}_{\mathcal{T}_j,\mathcal{R}_i}}{\sum_{k=1}^{m}\frac{P_{\mathcal{P}_k}}{d^{\alpha}_{\mathcal{P}_k,\mathcal{R}_i}} + \sum_{j'\neq j}^{n/2}\frac{SINR^{max}_{(\mathcal{T}_{j'},\mathcal{R}_{i'})}}{d^{\alpha}_{\mathcal{T}_{j'},\mathcal{R}_i}}} \tag{4.31}$$

The above equation is similar to $SINR^{max}_{(\mathcal{T}_j,\mathcal{R}_i)}$ but with an additional summation term in the denominator which accounts for the existence of other conditional candidates around $\mathcal{R}_i$. Thus a pair with high $SINR^{Rel}_{(\mathcal{T}_j,\mathcal{R}_i)}$ stands as a good candidate for increasing the system's capacity and hence a candidate power winner from the optimizer.

We show an illustrative example in Fig. 4.6 with 4 secondary transmitter-receiver pairs around their closest primary transceiver. The numerical values of the corresponding $SINR^{Rel}$ for each pair is analytically found and the corresponding power values from the global optimizer (optimization was done in Matlab) are also shown.

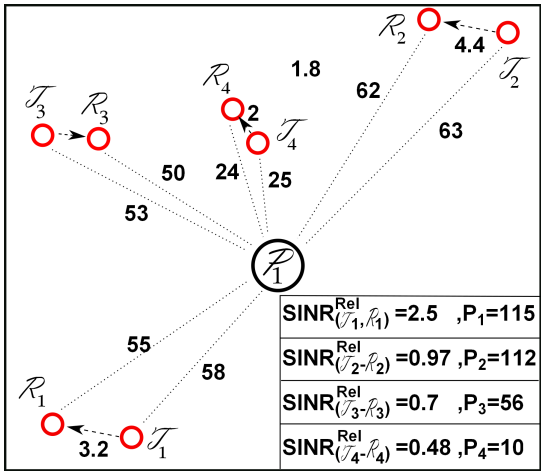


Figure 4.6 4 pairs around their closest primary transceiver with primary transceiver power=800 mW, $\gamma = 0.1$; the numbers on the links are the distances. The table shows $SINR^{Rel}$ values and their actual power allocations (in mW), a node with higher $SINR^{Rel}$ gets higher allocated power.

Inaccuracies in $SINR^{Rel}$: An important observation regarding the $SINR^{Rel}_{\mathcal{T}_j,\mathcal{R}_i}$ is, when:

$$\sum_{k=1}^{m} \frac{P_{\mathcal{P}_k}}{d^2_{\mathcal{P}_k,\mathcal{R}_i}} < 2 \times \sum_{j' \neq j}^{n/2} \frac{SINR^{max}_{(\mathcal{T}_{j'},\mathcal{R}_{i'})}}{d^{\alpha}_{\mathcal{T}_{j'},\mathcal{R}_i}}$$

the secondary interferers of $\mathcal{R}_i$ will add inaccuracies to its $SINR^{Rel}$ value. The effect of the inaccuracy appears in the pairs which have comparable $SINR^{Rel}$ values and are close to each other. Since their values are comparable and are in close proximity, the optimizer choose some of them as winners and leaves the rest unallocated, since they interfere each other highly (due to proximity) if all of them are given power.

When the power from the primaries dominate i.e., their interference is bigger than twice the interference from other secondaries, the effect of the un-allocation do not change the final result. When the primaries are not dominating, the un-allocation is reflected in the

summation that appears in the denominator of Eqn. (4.31). This is because, Eqn. (4.31) considers all pairs while calculating the $SINR^{Rel}$; while in reality it might so happen that not all the pairs with comparable $SINR^{Rel}$ are allocated power by the optimizer. This results in a scenario where some receivers are surrounded by more secondary interferers than the others, thus it affects the actual SINR and the pair's opportunity of winning power from the optimizer. Overall, this phenomenon adds inaccuracy to the ordering of the best pairs which we discuss next.

Ordered Set of Tx-Rx Pairs: We will show that for the same amount of power budget, a set that consists of pairs that have large $SINR^{Rel}$ yields better total capacity than a set having smaller $SINR^{Rel}$ values, given that the cardinalities of the two sets are equal. Question is: which subset of pairs maximizes the system capacity?

In a network with $n/2$ pairs, the corresponding power vector of the secondary users is $P = [P_1, P_2, ..., P_{n/2}]$. The ordered set $\mathcal{U} = \{U_1, U_2, \cdots, U_{n/2}\}$ is formed by re-arranging the elements of P in descending order ($U_{j-1} \geq U_j$) based on the $SINR^{Rel}$ values of the corresponding pairs. Next, a random subset $\mathcal{M}$ is formed by choosing pairs randomly from the set P, with $|\mathcal{M}| = |U|$. If the power budget is optimized over the elements of $\mathcal{M}$, it would result in some capacity. We argue: if any pair from $\mathcal{M}$ is to be replaced with a pair from $\mathcal{U}$ such that the $SINR^{Rel}_{U_i} > SINR^{Rel}_{M_i}$, the total capacity will increase. The question that arises is: what is the maximum achievable capacity using $L \leq n/2$ pairs with a given power budget?

Further investigation of the set $\mathcal{M}$ reveals that its current capacity can be improved by replacing the pair in $\mathcal{M}$ which has the lowest $SINR^{Rel}$ with U_1 (U_1 is the first element in $\mathcal{U}$). This replacement increases the total capacity. However, there might be still room for improvement because replacing the pair which currently has the smallest $SINR^{Rel}$ in $\mathcal{M}$, with U_2 (the elements of $\mathcal{U}$ are arranged in descending order) results in capacity improvement. The process is repeated until all L elements of $\mathcal{M}$ correspond to the first L elements from the set $\mathcal{U}$. At that point, it can be said that out of the $\binom{n/2}{L}$ possible sets, optimizing the set $\mathcal{M}$ yields the maximum expected capacity for the given power budget.

Subset Optimization: Here, we extend the idea of using the top pairs of $\mathcal{U}$ which was introduced in the previous section. The idea behind subset optimization is to achieve the highest capacity using only a subset of L pairs out of the $n/2$ pairs. Such scenarios are encountered when the convergence time for the $n/2$ pairs exceed the given time budget. In other words, due to time budget, it is not possible to optimize over the search space that is spanned by the $n/2$ dimensions of all the pairs. Instead, we can afford to optimize over a smaller search space of L dimensions only ($L < n/2$). So the idea is to optimize a partial set of length L whose elements have the highest $SINR^{Rel}$ values in $\mathcal{U}$ i.e., $\{U_1, \cdots, U_L\}$. We present a case study in Section 6.2.1.4.

4.2 QoS and Power Vector Evaluation

We outlined in Section 2.2, that there has been lots of work on power control and QoS metrics. While the focus of these works have been mainly on obtaining the optimized power vector, evaluating the QoS metrics of these power vectors were left open. Such evaluation becomes even more critical for "low cost" DSA devices [92]. Importance of such devices/modules emerges as they open new horizons for building low cost DSA devices. Of course, the use of such devices is accompanied by the cost vs. processing dilemma i.e., as the price goes down so does the processing power. An issue that has been ignored is how to manage the computations while keeping the cost low.

To that end, we propose a predictive model that takes into consideration the power levels of a set of secondary transmitters and provides the expected performance of the system. We show how the power vector which is the result of a power control scheme is nothing but a point in a K-dimensional space. We consider the power control scheme as a black box. Using vector quantization, we partition the space into different regions that reveal what the allowed power level of each transmitter is. Thus, by knowing the power vector and recognizing which partition it belongs to, we are able to determine the current state of the system including the expected performance in $O(\log\ N)$ time.

4.2.1 K-dimensional QoS Space

We consider a secondary network where secondary transmitters and receivers are scattered randomly over the region of interest in the presence of primaries. We consider a generic power control algorithm which provides the transmit power levels for each transmitter. It can be noted that the output power vector is a function of time due to noise, fading, and primary transmit power variations. The objective of the power control algorithm is to maximize a given QoS-metric for example, capacity, spectrum usage, primary service degradation, interference at the primary, etc.

We consider the power levels for the K secondary transmitters at time t and represent them by a K-dimensional vector

$$\mathbb{P}_t = [P_1, P_2, ...P_K]_t$$

where P_i is the transmit power of transmitter i. Based on the instantaneous values of P_i's at any time t, $\mathbb{P}_t$ can be thought of as a point in the K-dimensional continuous space at that time. Due to the dynamic channel conditions and primary activities, the values of P_i's continuously change– the optimal value of which is determined by the employed power control scheme. Thus, $\mathbb{P}_t$ can be seen as a loci of a point that moves in the n-dimensional space. Thus the power vector determines the current system status and provides insights into the level of QoS being attained by the system. It can be noted that the power for the non-transmitting nodes can be set to 0.

For every P_i, there are some range of values that indicate i) normal operating conditions, ii) abnormal conditions, and iii) unfeasible conditions. For example, transmitting at 1 mW could be normal, transmitting at 10 mW could cause harmful interference to a primary, and a power level of 100 mW might not be possible due to hardware constraints. Just as we can identify various ranges for P_i, we can think of various regions in the K-dimensional space that offer different service quality. As a matter of fact, every point in this space signifies a level of performance or the QoS offered by the system. Thus, we associate every point with a *QoS index* that represents the current state of the system. As mentioned earlier, our use of

the term QoS is generic; it can refer to a variety of attributes like capacity, spectrum usage, primary network degradation, etc.

4.2.2 QoS Space Partitioning

Since vector $\mathbb{P}_t$ can take any real value, we get a real-valued space. Dealing with such a space with infinitely many QoS indices is not only cumbersome but also computationally intractable. A better approximation would be to represent nearby points by a *representative* point with its corresponding QoS index. In this work, we apply our technique to capacity, i.e., the QoS index is the capacity produced by the representative's $\mathbb{P}_t$ value. These neighboring points actually define a region with a strict boundary. Now the question that arises is: how to obtain the non-overlapping or disjoint partitions– the union of which spans the entire region?

The problem of finding the disjoint regions boils down to partitioning the QoS space into a finite number of regions, each associated with the QoS index of the region's representative point. The QoS of every point within a region would be represented by the QoS index of that region. So, instead of dealing with infinitely many indices, we will have a finite number of indices representing different regions in space. This scenario is illustrated in figure 4.7, where a 2-dimensional space is spanned by disjoint regions. This diagram is commonly referred to as the *Voronoi diagram* [93]. We show only the first quadrant because the power levels (P_i's) can only be non-negative. Each of the regions will have a *representative* point,

which is usually the *centroid* (shown by the solid dots), and any point in that region maps to that representative point. It is to be noted that, theoretically the QoS space is unbounded since P_i is unbounded, for $1 \leq i \leq K$. Optimal partitioning of the unbounded space with a finite number of bounded regions is a hard problem. Hence there exists some unbounded regions as shown in figure 4.7. However from a practical viewpoint, all transmitters have limited capability, thereby making the QoS space bounded.

Partitioning the QoS space is analogous to *space quantization*, where the space is partitioned into N quantization regions, and every point in the space belongs to one of the regions. If we can identify the region in which the system is currently operating then we have an idea about the QoS as experienced by the system. In other words, the power vector representing the system status maps to one of the representative points in the K-dimensional QoS space.

It is not necessary that the QoS indices of all the N partitions are distinct, since two different partitions might offer the same QoS and hence could have identical QoS indices. An exact quantification of the QoS index is non-trivial because it is difficult to find a single value that would capture all the attributes contributing towards the QoS.

4.2.2.1 Mapping Power Instances

As pointed out earlier, the dynamism of the system is manifested by the ever-changing power vector $\mathbb{P}$ which is mainly due to the channel characteristics and primary activities. Every

instance of the power vector $\mathbb{P}$ can be mapped to one of the N representative points in space. Each time a new instance of $\mathbb{P}$ is encountered, its relationship with the previous instance is examined in order to find the region it belongs to, or more precisely the representative point it maps to.

One approach to finding the *target* (representative) vectors could be to retrieve *similar* instances from memory and classify them based on the previous classification. But the problem is that the target points for the new instance may not be the same as the one obtained from the previous classification. This is possible if the retrieved similar instances were mapped to different target points. Moreover, there is a huge computation associated with the classification.

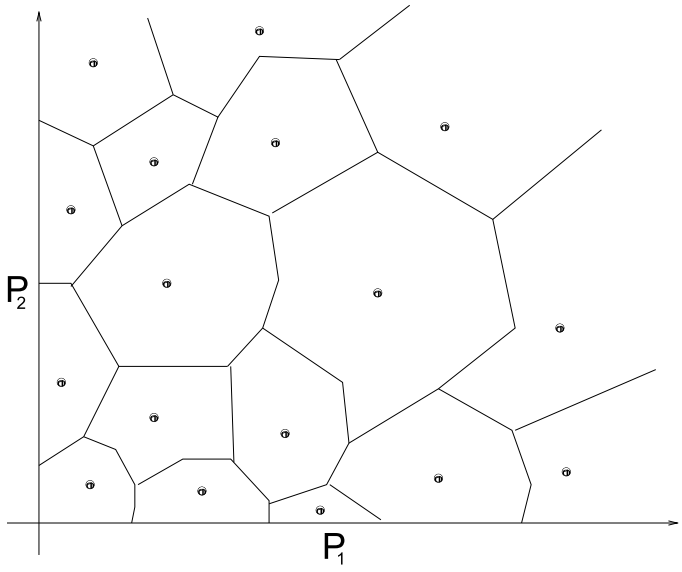


Figure 4.7 Partitions with their centroids

One technique to avoid this cumbersome computation is to use the *1*-Nearest Neighbor algorithm which is a modified version of the K-Nearest Neighbor algorithm [93]. The K-Nearest Neighbor algorithm assumes that all instances are mapped to points in the K-dimensional space. The nearest neighbors are usually defined in terms of Euclidean distance. For example, if an instance is $\mathbb{P} = [P_1, P_2, \cdots, P_K]$ then the *distortion* d_j is simply the Euclidean distance from $\mathbb{P}$ to the target point $M_j = [M_{j1}, M_{j2}, \cdots, M_{jK}]$, where $1 \leq j \leq N$, is given by

$$d_j = \sqrt{\sum_{i=1}^{i=K} (P_i - M_{ji})^2}.$$

4.2.2.2 Design Objectives

We would not be interested in uniformly partitioning the space, as done in uniform quantization, since the power vectors are not uniformly distributed in space. There is a density variation of the power vectors in the sense that the system might tend to operate in a particular region more often than others. Moreover, uniform quantization does not yield an optimal solution with respect to the global distortion. Our goal is to find the target point which yields minimum distortion and also minimizes the search time for that target point.

1. Minimizing Distortion: The values of d_j's, for $1 \leq j \leq N$, give the distances between the new instance and all the N target points. The nearest target point can be obtained by finding the minimum of d_j's for all j. Thus,

$$M_j \leftarrow \mathbb{P} \text{ for which } d_j \text{ is minimum.}$$

Thus, the new instance of $\mathbb{P}$ maps to the nearest target point M_j and obtains the QoS index of M_j. Of course, there is an error introduced due to the mapping– the larger the number of target points, the smaller the distortion. The average distortion is a good measure of the quantizer since we aim to minimize the average distortion.

2. Minimizing Search Time: If a linear search is employed to find the nearest target point, the required time would be $O(N)$. For large values of N, though the average distortion will be less, the search for the nearest neighbor would be high for any real-time decision. A technique for efficiently indexing the space is a significant practical issue in minimizing the computation required at the query time. We borrow concepts from tree structured vector quantization (TSVQ) that takes $O(log\ N)$ time [94].

The question still remains, how to optimally partition the space and identify the different regions, each of which would be represented by one QoS index. In the following subsection, we show how to construct the representative vectors so as to bring down the search time and also keep the global distortion minimum. We adopt a technique called *vector quantization* [94], which is a generalized version of scalar quantization.

4.2.3 Quantizing the QoS Space

Before applying vector quantization to partition the QoS space, let us first discuss scalar quantization for the sake of completeness and better understanding. In *scalar quantization*, a 1-dimensional space is partitioned into multiple regions and any point in a particular region is mapped to the representative point of that region. More precisely, an N-point scalar quantizer $\mathcal{Q}$ is a mapping function such that $\mathcal{Q} : \mathcal{R} \rightarrow \mathcal{C}$ where $\mathcal{R}$ is the real number line and

$$\mathcal{C} \equiv \{M_1, M_2, M_3, \cdots, M_N\} \subset \mathcal{R}.$$

The output set $\mathcal{C}$ is popularly called the *codebook* of size $|\mathcal{C}| = N$. We will use the terms $|\mathcal{C}|$ and N synonymously to refer to the size of the codebook. The output values, M_i, are also referred to as *reproduction values*. Associated with every N point quantizer is a *partition* of real line $\mathcal{R}$ into N *cells* or *regions* $\mathcal{R}_i$, for $i = 1, 2, \cdots, N$. The ith region is given by

$$\mathcal{R}_i = \{x \in \mathcal{R} : \mathcal{Q}(x) = M_i\} \equiv \mathcal{Q}^{-1}(M_i),$$

the inverse image of M_i under $\mathcal{Q}$. Thus, we see that a quantizer $\mathcal{Q}$ can be completely described by

$$\mathcal{Q} = \{M_i, R_i;\ i = 1, 2, \cdots, N\}$$

in terms of its reproduction points $\{M_i; i = 1, 2, \cdots, N\}$ and the corresponding regions $\{\mathcal{R}_i; i = 1, 2, \cdots, N\}$.

Vector Quantization (VQ) is a generalization of scalar quantization where an ordered set of real numbers is quantized. A K-dimensional vector quantizer $\mathcal{Q}$ is a mapping from a point (a vector) in the K-dimensional Euclidean space, $\mathcal{R}^K$, into a finite set $\mathcal{C}$ containing N reproduction points. These reproductions points are called the *codewords*. Thus,

$$\mathcal{Q} : \mathcal{R}^K \rightarrow \mathcal{C}.$$

As in scalar quantization, each of the N codewords is associated with a region $\mathcal{R}_i$, such that

$$\mathcal{R}_i = \{x \in \mathcal{R}^K : \ \mathcal{Q}(x) = M_i\}.$$

The set $\mathcal{C}$ is also called the *codebook* which has size N, meaning it has N distinct vectors. The goodness of a codebook is measured by the *distortion* which is defined as the non-negative cost $d(x, \hat{x})$ associated with quantizing any input vector x with a reproduction vector $\hat{x}$. The codebook of a VQ is said to be *optimal* if it minimizes the average distortion which quantifies the performance of the system.

Since neither the input instance (power vector) nor its probability distribution is known beforehand, the design of the codebook is heavily dependent on the *probability density function* (pdf) of the input vectors. A reasonable approach is to take long sequences of training vectors and estimate the average distortion. It is difficult to come up with a pdf which would replicate the actual power vectors during the normal course of operation of the system. In that case, the only option is to use real data gathered from the system itself and use them as training vectors. If the input training vectors are stationary and ergodic, the resulting average distortion on future data should yield approximately the same

distortions [95]. If we deal with *sufficiently long* sequences of training vectors, then the performance of the obtained codebook on new and real data would be the same. Given the training vectors, there are several approaches for the construction of codebook [94, 95, 96]. We choose the *splitting method* (also known as the LBG algorithm [97]). Though improved versions have been proposed by [98, 99, 100], we choose the original method due to its simplicity.

4.2.3.1 Finding the Codewords

In this iterative method, the size of the codebook grows from 1 to the desired value N. Given the set of training vectors, if we are to have a codebook of size $N = 1$, then the reproduction vector would be the centroid of these training vectors. This is illustrated in figure 4.8(a) where the solid dot M_1 denotes the centroid of the training vectors represented as hollow dots. It also means that we have only one partition R_1. For the sake of convenience, a 2-dimensional space is considered, which means we have two transmitters operating with power levels P_1 and P_2 respectively. The centroid is the only point in the space whose sum of the Euclidean distances to all the training vectors is minimum. (The position of the centroid M_1 as shown in figure 4.8(a) might not be accurate as it is for demonstration purpose only.) The following question might arise due to the discrete nature of the training vectors. It was previously argued that the power vector has a continuous motion. Strictly speaking the training vectors should have been the locus of the power vector. But the locus

can be sampled periodically at a certain rate to obtain those discrete points. In other words, the hollow dots represent the snapshots at equals interval of time.

We start with just one entry in the codebook which is the centroid, say M_1. This codeword is then split into two codewords, M_1 and $M_1 + \epsilon$, where ϵ is a vector of small Euclidean norm. An *iterative clustering* algorithm can now be used to find the optimal positions of these two codewords. Figure 4.8(b) shows the scenario with two codewords corresponding to the two partitions R_1 and R_2. On splitting these two codewords into four and applying the clustering algorithm iteratively, we obtain the four centroids along with the four partitions as shown in figure 4.8(c). Note that the partitions R_1 and R_2 are bounded, whereas partitions R_3 and R_4 are unbounded. At the next step, eight partitions would be obtained, the figure for which is not shown.

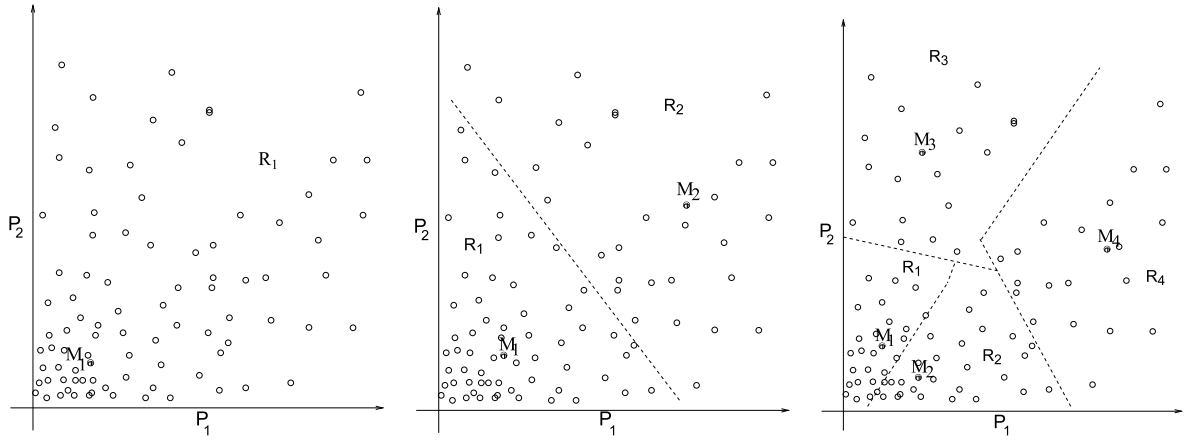


Figure 4.8 a) One codeword and one partition; b) Two codewords and two partitions; c) Four codewords and four partitions.

4.2.3.2 Choice of N

The main goal of a quantizer design is to find the codewords and the partitions such that the average distortion is minimized for a fixed number of codewords. The minimum distortion also gives a measure of the *resolution* of the quantizer. This minimum distortion can be used to back-calculate the number of codewords which would be necessary. More explicitly, the average (mean square) distortion D is given by

$$D = \sum_{i=1}^{N} \int_{\mathcal{R}_i} (x - M_i) f_X(x) dx, \tag{4.32}$$

where M_i is the codeword in the region $\mathcal{R}_i$ and $f_X(x)$ is the pdf of the random variable X. We sum the distortions in all the N regions and integrate within each region $\mathcal{R}_i$ because of the continuous space, thus capturing the spatial-temporal aspect of the power vector. Thus, if the tolerable distortion is given, the number of codewords N can be obtained.

4.2.3.3 Iterative Clustering

In iterative clustering, all the training vectors are made to map on to the nearer of the two codewords y_0 and $y_0 + \epsilon$. As a result, two clusters will emerge as some of the training vectors will map to y_0 and the others will map to $y_0 + \epsilon$. The centroids of the two clusters will be found and the two codewords, y_0 and $y_0 + \epsilon$, will be updated with the centroids' position (i.e., the codewords are displaced to decrease the distortion). Clustering will be performed

again on these two codewords. (Note that it is not necessary for a training vector in one cluster to be mapped on to the same cluster after the displacement of the codewords.) The clustering and the displacement of the codewords are done iteratively till the displacements become negligible. It has been shown in [95] that for a finite set of training vectors, the splitting algorithm always produces a sequence of vector quantizers whose average distortion converges in a finite number of iterations. The final positions of y_0 and $y_0 + \epsilon$ will be the entries in the codebook of size 2. To obtain codebook of higher orders, these two codewords are again split into four and their optimal positions are found. This process is continued till the desired size (N) of the codebook is obtained.

4.2.3.4 Search in $O(logN)$ Time

Tree structured VQ, denoted as TSVQ, is a technique to reduce the search complexity in VQ. Due to the nature of the splitting algorithm, the codebook can be stored in as tree structure which can reduce the search time. Such a tree structured VQ is a natural byproduct of the splitting algorithm. As opposed to linear search which takes $O(N)$ time, finding the nearest neighbor using a tree takes $O(logN)$ time, if there are N entries in the codebook. In TSVQ, a binary search starts with comparing the SINR vector instance with the two codewords which were the outcome of the codebook generation process for $N = 2$. Note that these two codewords are not part of the codewords in the final codebook. This is just one of the intermediate stages in the codebook generation process. Figure 4.9 gives an example of a

TSVQ for $N = 8$ with the intermediate codebooks of size 2 and 4 are also stored for the purpose of search. The final 8 codewords are marked as M_1 through M_8. of size 8. Of course, the process of storing the intermediate codebooks is recursive which can generate codebook of any size with is an exponent of 2.

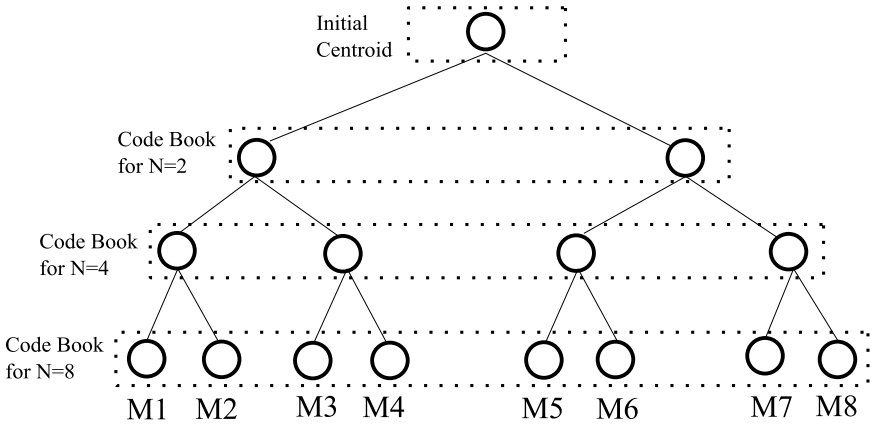


Figure 4.9 Codebook storage and lookup

4.2.3.5 Updating the Codebook

So far we have considered an off-line and one-time construction of the codebook based on the training vectors. That is, the codebook is constructed once and all real signal vectors are coded based on that. But it might so happen that the signal vectors exhibit non-stationary behavior. A natural way to adapt the quantizer is to dynamically adapt the codewords in the codebook based on acquiring updated information about the power vectors. In this manner,

improved coding performance is possible if the codebook can somehow adapt its codewords to suit the local-stationarity of the actual power vectors.

4.2.3.6 An Illustration: Putting It All Together

Let us give an illustration as the power vector $\mathbb{P}$ moves in figure 4.10. This is the same two-dimensional QoS space as shown in figure 4.7. The solid dots represent the representative points of the respective partitions. Let $\mathbb{P}(t)$ be the position of the power vector at time t belonging to the partition 1 with representative point M_1. We consider two possible scenarios at time $(t + \Delta t)$ as follows. $\mathbb{P}_1(t + \Delta t)$ undergoes a displacement of $\vec{\mathcal{D}}_1$ and remains in the same partition 1. Hence it maps to the same representative point M_1 as for $\mathbb{P}_1(t)$, yielding the same QoS index. Whereas, $\mathbb{P}_2(t + \Delta t)$ undergoes a displacement of $\vec{\mathcal{D}}_2$ but moves to a different partition 2, and maps to a different representative point M_2.

4.2.4 Proof of Concept

To provide a proof of concept, we simulate a simple DSA network with two transmitters. Thus, the output of the power control algorithm is a vector $\mathbb{P} = [P_1, P_2]$, where P_1 and P_2 are the transmit powers for transmitter 1 and 2 respectively. (Having 2 transmitters will allow us to pictorially illustrate the concept of QoS space partitioning.)

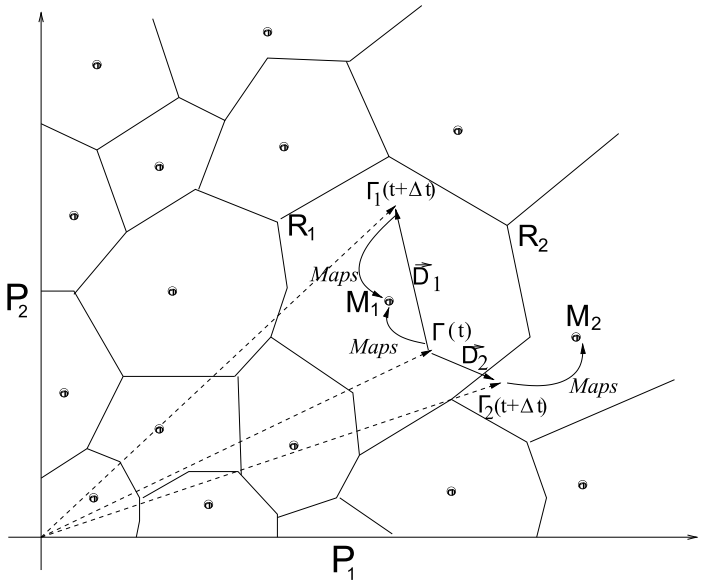
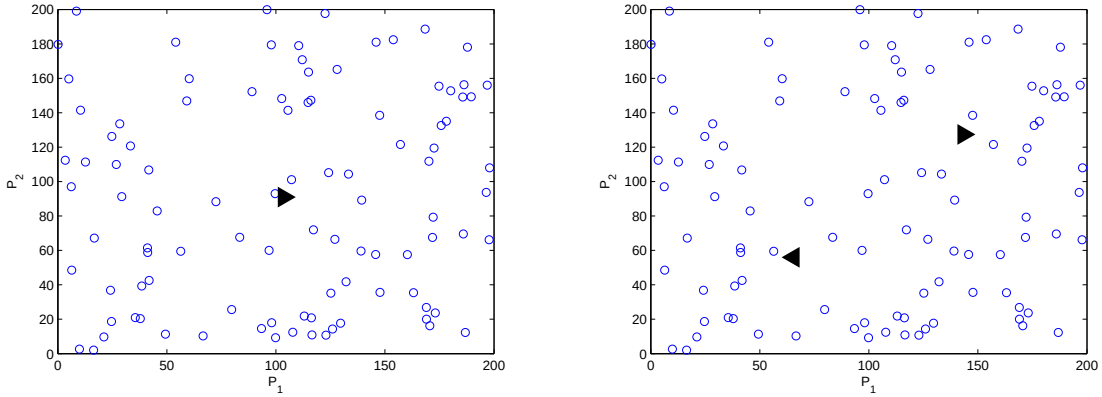


Figure 4.10 Displacement vector

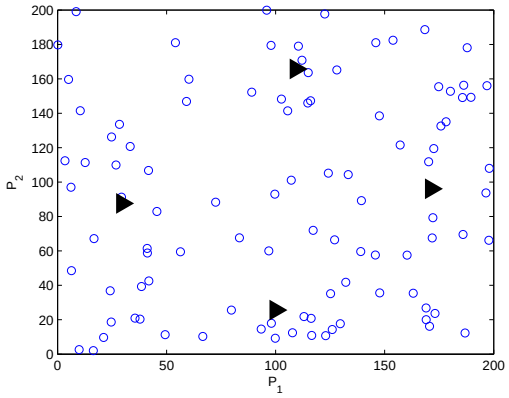
The two transmitters along with their respective receivers are randomly placed over an area of 100 X 100 with the primary being at (50, 50). We randomly generate 100 instances of [P_1, P_2] pairs and partition them using the LBG algorithm. P_1 and P_2 were generated uniformly randomly between 0 and 200 mW. The first phase results in 1 centroid (black solid triangle) as shown in Fig. 4.11(a). Continuing with partitioning and clustering we obtain 2, 4, and 8 centroids as shown in Figs. 4.11(b)-4.11(d). We do not continue further.

As for a specific QoS metric, we consider the capacity of the system which is nothing but the sum of the capacities of all the transmitter-receiver pairs. For K transmitter-receiver pairs, we evaluate the Shannon capacity, C, as:

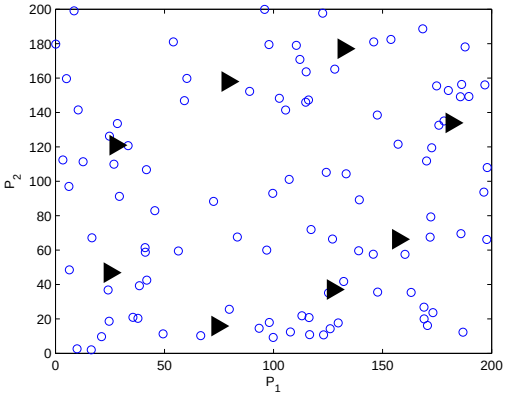
$$C = B \sum_{i=1}^{K} \log_2 \left(\frac{\frac{P_i}{{d_{ii}}^{\alpha}}}{N_0 + \frac{P_p}{d_{pi}^{\alpha}} + \sum_{j=1, j \neq i}^{K} \frac{P_j}{d_{ij}^{\alpha}}} \right) \tag{4.33}$$



(a) (b)



(c)



(d)

Figure 4.11 a) 1 centroid; b) 2 centroids; c) 4 centroids; d) 8 centroids

where B is the bandwidth, N_0 is the noise, d_{ij} is the distance between the ith transmitter and the jth receiver, d_{pi} is the distance between the primary transmitter and the ith receiver, P_p is the power of the primary, and α is the path loss exponent.

For illustration and better pictorial representation, we consider $K = 2$, i.e., two transmitter-receiver pairs. Thus, equation (4.33) reduces to:

$$C = B\log_2\left(\frac{\frac{P_1}{{d_{11}}^\alpha}}{N_0 + \frac{P_p}{d_{p1}^\alpha} + \frac{P_2}{d_{21}^\alpha}}\right) + B\log_2\left(\frac{\frac{P_2}{{d_{22}}^\alpha}}{N_0 + \frac{P_p}{d_{p2}^\alpha} + \frac{P_1}{d_{12}^\alpha}}\right) \tag{4.34}$$

For each instance of $\mathbb{P}$ as shown in Fig. 4.11(a)-4.11(d), we plot the corresponding capacity in Fig 4.12 which is the 2-dimensional QoS (capacity) space. We partition this capacity space into 8 regions and find the 8 centroids as shown in Fig. 4.13(a). Again, selection of 8 regions is for graphical demonstration purpose only. (It can be noted that the projections of the 8 points in Fig. 4.13(a) onto the $P_1 - P_2$ plane are the same as the 8 centroids in Fig. 4.11(d).) Once these 8 entries for the codebook are created; we simply consult the codebook to find the closest centroid for a new power vector.

Search complexity: The corresponding capacity can be found by a fast look-up and comparison with the pre-computed codebook (centroids) and *not* by computing Eqn. (4.33) which is tedious for large number of transmitter-receiver pairs. Though a linear search for the nearest centroid takes $O(N)$ time, we use the tree structured vector quantization that takes $O(log\ N)$ time [94].

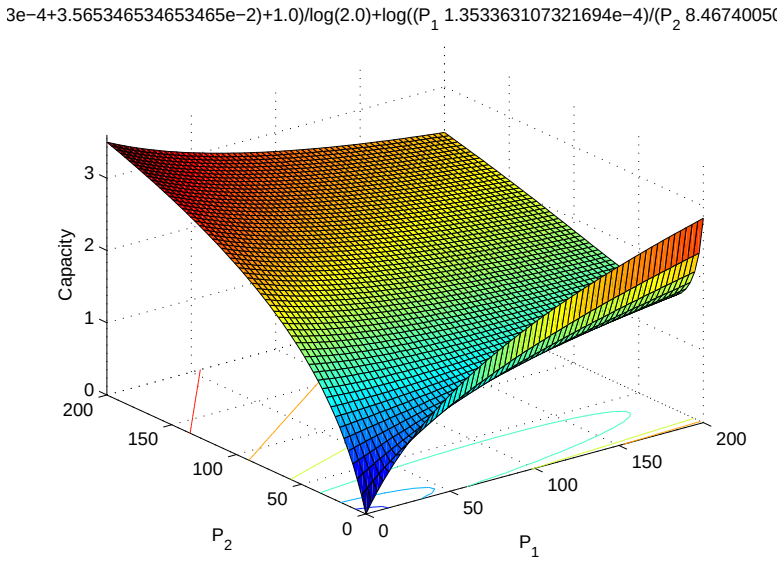
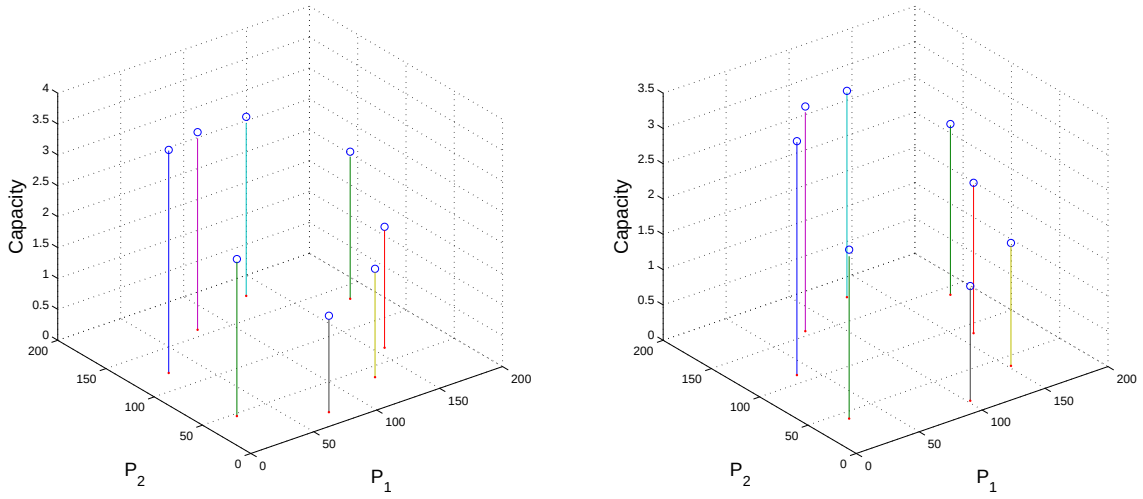


Figure 4.12 Capacity as the objective function



(a) (b)

Figure 4.13 a) 8 centroids in the capacity space with 100 data points; b) Updated 8 centroids in the capacity space with 125 data points.

Codebook update: As mentioned earlier, we used 100 data points (i.e., power vectors) to generate the codebook with 8 entries. As we get more data points, we are able to update the codebook. With additional 25 data points, we update the codebook as shown in Fig. 4.13(b). Note that, the initial centroids have moved slightly due to the new data points.

It can be noted that the selection of the data points influences the codebook. Therefore, we used a training set that produces most of the possibilities given the restrictions on the upper bound on the transmit power and the localization of the nodes. Should there be any deviation in the statistical nature of the data points, a continuous update process performed off-line time will keep the codebook updated.

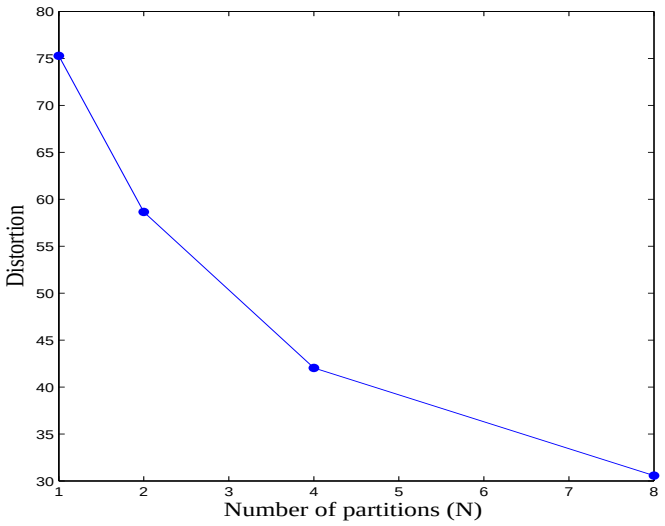


Figure 4.14 Distortion with increasing no. of partitions

Distortion Vs. N: Distortion, as was given in Eqn. (4.32), is a measure of the goodness of a codebook. Of course, there is a trade-off between the number of partitions (i.e., entries in the

codebook) and the search time. For the same distribution of P_1 and P_2 (i.e., between 0 and 200) we find the distortion. As expected, the distortion decreases with increasing number of partitions (N) as shown in Fig. 4.14.

4.2.4.1 Reverse Lookup

As mentioned earlier, that although the proposed method does not perform power control, it can still be used to bypass the complex resource optimization by performing a reverse-lookup as follows. If a specific network QoS (say Q_{target}) is desired then the codebook can be looked up in $\mathcal{O}(\log N)$ time to find the centroid with the closest QoS value (say, $Q_{closest}$) to Q_{target}. Once $Q_{closest}$ is determined, its corresponding power vector can be assigned to the secondary users to achieve a QoS value that is closest to the target QoS without the need to perform optimizations.

Moreover, the codebook can be constructed to include all the different resources such as time frames, frequencies, codes, that are associated with each power vector. Following such an approach, the reverse look can help decide on not only power but also other types of resources that are associated with $Q_{closest}$.

4.2.4.2 Impact On The Primaries

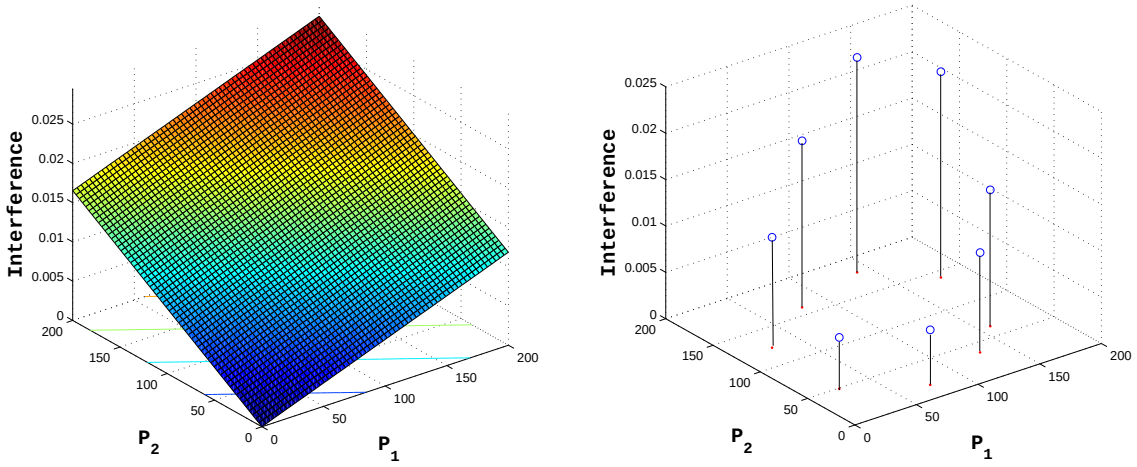
Since the secondaries use the same channel as the primary, the primary will get interfered from all secondaries– the magnitude of which will depend on their mutual distance and the path loss exponent. The total interference perceived at the primary, I_p, is given by

$$I_p = \sum_{i=1}^{K} P_i / d_{i,p}^{\alpha} \tag{4.35}$$

where $d_{i,p}$ is the distance from the ith secondary transmitter to the primary.

An Illustrative Example with 2 Secondary Pairs: Let us consider a primary receiver located at $(0, 0)$ and two secondary transmitters at distances of 110 and 130 Meters respectively. The transmit power ranges from 0 to 200 mW. For each instance of $\mathbb{P}$, we plot the corresponding interference at the primary for all values of P_1 and P_2 creating the *Interference Space* as shown in Fig. 4.15(a).

Following the same approach in Section V (in creating the centroids for the capacity), we partition the space into 8 regions and find the 8 centroids as shown in Fig. 4.15(b). These 8 centroids will be used to map the new values of $\mathbb{P}$ to find the expected interference on the primary.



(a) (b)

Figure 4.15 (a) Interference space of the primary, (b) Partitions of the interference space with their centroids

4.3 Summary

In this chapter, we showed how to bound the maximum achievable capacity of a finite secondary DSA network operating in the underlay mode using the SINR model. We also presented the concept of dead pairs and used it to reduce the optimizer's search space without comprising the optimality. We also presented $SINR^{Rel}$ and illustrated how to use it to obtain sub-optimal solution or as well as to further reduce the optimizer's search space.

Next, we presented a vector quantization based QoS evaluation scheme where the K-dimensional QoS space is partitioned and each partition has a representative point which can be looked up in $\mathcal{O}(\log N)$ or $\mathcal{O}(N)$ time for the expected QoS metric.

CHAPTER 5: DESIGN AND IMPLEMENTATION OF LOW COST DSA SYSTEM

This chapter presents the design and implementation of a low-cost DSA system. We start by outlining our design philosophy and present the system architecture with its components. We discuss in detail the design of each component. We also present the details of the hardware and software implementation.

5.1 Design Philosophy

Although many techniques exist for spectrum sensing like matched filer, energy detection (ED), cyclostationary, etc [101, 102, 103]; energy detection stands out as one of the optimum methods as it is simple and does not require any prior knowledge about the primary user(s) [104]. Simply stated, ED revolves around measuring the amount of energy on a specific bandwidth. In traditional DSA systems, heavy duty hardware devices (SDRs, spectrum analyzers) are employed for this simple task which is an overkill. Also, the cost of these devices becomes a bottleneck for large scale deployment. We argue that these light-weight sensing tasks can potentially be done on relatively inexpensive and computationally light devices.

Our approach is based on the fact that recent years have witnessed a proliferation of low cost off-the-shelf software configurable RF (SCRF) chips. Popular examples are the CC2500, CC1100, CC2550 by Texas Instruments [105] and RFM22, RFM23, and RFM69 by HopeRF [106]. These devices have built-in capability to record RSSI values (digital or analog)– a feature which we believe has been overlooked for its potential for spectrum sensing, particularly when combined with the energy detection algorithms like localization algorithm double-thresholding (LAD), forward consecutive mean excision (FCME), etc [107, 108]. Our design philosophy is based on i) *harnessing the inexpensive commercial off-the-shelf SCRF chips* that offer this RSSI feature, instead of traditionally used SDRs allowing for a low-cost solution to sensing, scalability, and cost, and ii) *realizing flexible system configurablity* via user-friendly programming interface. Though the spectrum analyzers and SDRs are far more capable than the SCRF chips for spectral analysis, however for ED, such features are irrelevant. Most importantly, the SCRF chips sold for $3-$4 are orders of magnitude economical than SDRs and spectrum analyzers.

5.2 System Architecture

The proposed DSA system architecture, which has also been implemented, is shown in Fig. 5.1. The overall system architecture consists of three primary components– each of which has multiple sub-components. The primary components are: i) the RSSI sensing network, ii) the DSA server, and iii) the service engine. The sensing network captures the

RSSI readings and streams them real-time to the DSA server where the data is received, parsed, and stored in a database. The RSSI records from the database are made accessible to any end-user by the service engine which offers a wide variety of services.

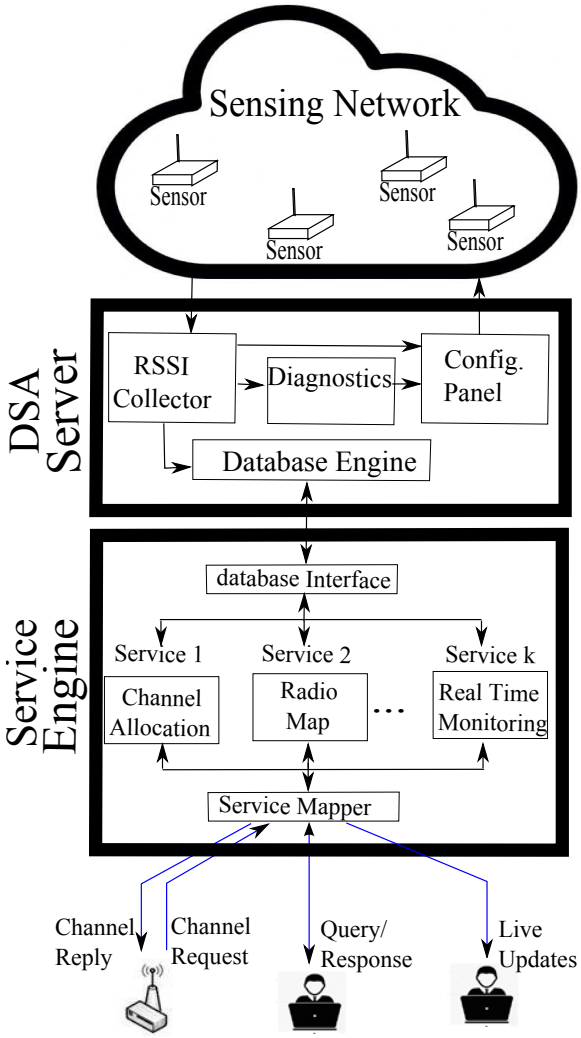


Figure 5.1 System Architecture.

The sensing network consists of a variety of low-cost spectrum sensors which are capable of streaming real-time RSSI measurements (wired or wirelessly) to the DSA server. The RSSI collector in the DSA server continuously receives the streams of data from multiple

sensors and passes them to the database engine which creates a record (that includes time-stamps, sensor ID, sensor location) for each reading. The configuration panel module allows changes to the radio and network parameters for the sensing network based on the real-time performance diagnostics. The service engine facilitates various end-user services like channel allocation to DSA nodes, radio map environment (REM) construction, real-time updates on spectrum occupancy on various bands, etc.

5.3 RSSI sensing network

The sensing network consists of spectrum sensors which can be wired or wireless. We must first understand the building blocks of the sensor nodes.

5.3.1 Spectrum Sensors

A spectrum sensor unit/node primarily consists of two main components: the RF-front end and the controller. Depending on how the senor would be connected to the rest of the network, there could an additional Ethernet interface. A block diagram along with what has been actually implemented is shown in Fig. 5.2. We emphasize that the design is not based on any specific component. These components are treated as generic building blocks. Our goal is to provide a guideline on how to capitalize on the off-the-shelf SCRF chips and controllers for making low-cost spectrum sensors.

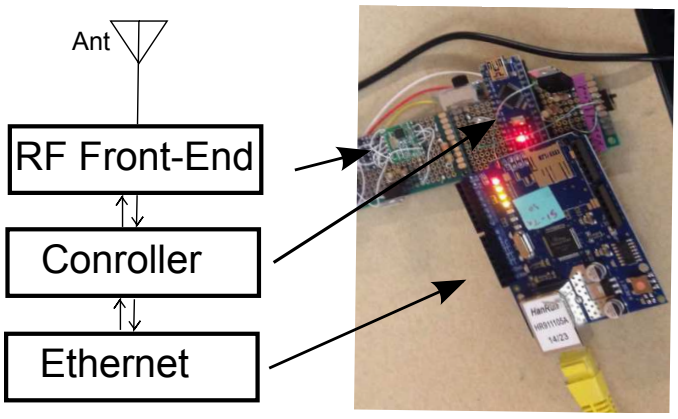


Figure 5.2 Block diagram of sensor node (left) with correspondence to an implemented unit (right).

5.3.1.1 RF front-end

The RF front-end is responsible for measuring the RSSI values on the target bands. The RF front-end makes use of the RSSI reporting feature that is available in the off-the-shelf SCRF chips. As discussed earlier, these SCRF chips can be configured as transmitters (Tx) or receivers (Rx) in addition to configurable modulation schemes, bandwidth, and data rates. The features of any specific SCRF chip dictate the RF properties and capabilities of the sensor. For example if the SCRF chip uses an analog technology with maximum bandwidth of 500 KHz, then the sensor will have an analog front end with 500 KHz as the maximum bandwidth. If needed, the RF front-end can also be used to transmit the measured RSSI values over a radio channel as will be shown later in Section 5.3.

5.3.1.2 Controller

In order to configure and control the SCRF chips, a controller is needed– the interface of which depends on the SCRF chip in question. The serial peripheral interface (SPI) is a popular choice for TI and HopeRF chips. The controller instructs the SCRF chip with the RF configuration parameters (target frequency, bandwidth, scanning rate and resolution) based on the user inputs. The RSSI values from the RF front-end are then read by the controller. Depending on the network architecture, the controller either i) instructs the Ethernet interface to stream the RSSI values to the DSA server using some real-time streaming protocol, or ii) reconfigures the RF-front end as a transmitter to transmit the RSSI value over a radio channel.

5.3.1.3 Ethernet Interface

This optional interface can be used to make use of an existing network infrastructure to stream the RSSI values, thus relieving the RF-front end for sensing only. It can be noted that, the Ethernet circuitry could be integrated with the controller or it could be external to the controller.

5.3.2 Network Architecture

Though the Ethernet-enabled sensors can stream the RSSI values directly to the DSA server, the wireless sensors cannot do so as the DSA server is not wireless enabled. A wireless enabled DSA server does not help because of i) additional cost due to specialized hardware, and ii) reduced sensing area as the sensors will have to be able to directly communicate with the DSA server iii) lack of support for multiple frequencies.

In order for the *wireless* sensors to send their data to the DSA server, the sensors must send the data to a wireless hub which will then send the data to the DSA server via Ethernet as shown in Fig. 5.3. The figure also shows the Ethernet-enabled sensors that stream directly to the DSA server over the Internet. Apart from the hub and the sensors streaming data over the Internet, they could also be directly connected to the DSA server. Both configurations are illustrated in Fig. 5.3.

5.3.3 Network Operations

Let us now elaborate on how the various network components communicate over RF and Ethernet.

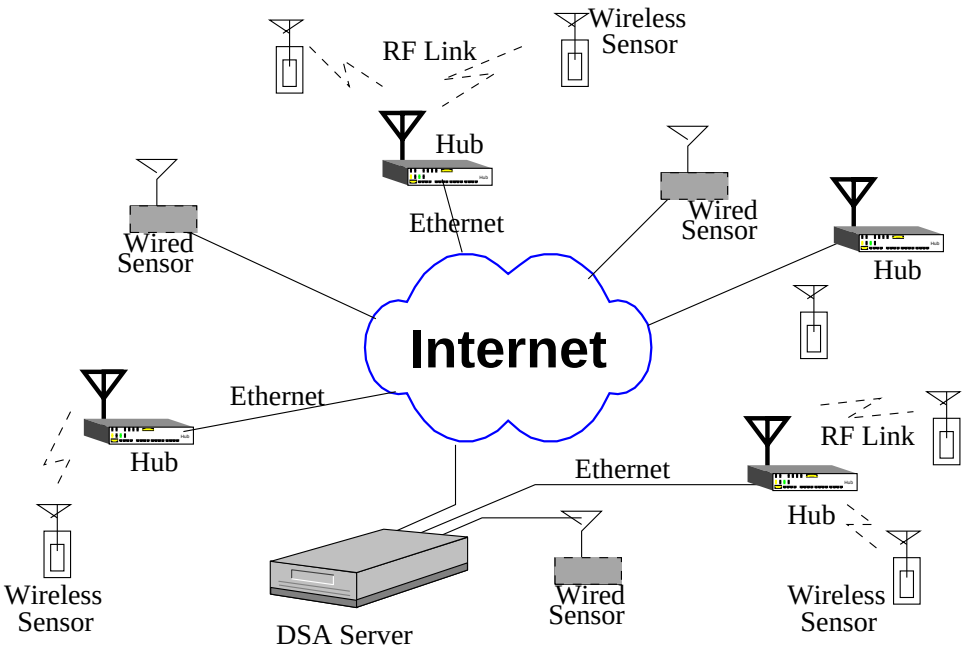


Figure 5.3 Overall network architecture showing how wireless sensors, hubs, and Ethernet-enabled sensors connect to the DSA server.

5.3.3.1 Wired Sensors

Recall, the wired sensors have an RF-front end, a controller, and an Ethernet interface. On power-on, each sensor contacts the DSA-server over Ethernet to obtain the list of frequencies that need to be scanned including bandwidth, sweep time, and scan resolution (i.e., scan-list). Once the scan-list is obtained, the controller cycles through the list one frequency at a time and instructs the RF front-end to measure the RSSI on each frequency. The controller then fetches the RSSI value from the RF front-end and passes it to the Ethernet interface where the controller instructs the Ethernet interface to build a UDP packet to be streamed to the DSA server. The payload of the UDP packets can carry: i) single RSSI value or ii) multiple RSSI values– the decision of which depends on the time critically of

RSSI reporting. Apart from the RSSI value(s), the payload contains other information such as `nodeID, frequency, sensor coordinates`.

5.3.3.2 Wireless Sensors

The wireless sensors use the *same* RF front-end for sensing and for transmitting the sensed data to the hub. On a broadcast channel, the hub instructs each wireless sensor to scan a specific set of frequencies. It also manages the channels (i.e., uplink frequency and the transmit time slot) for each sensor. This switching between sensing and transmitting by the RF front-end is achieved by the controller. It starts with the controller instructing the RF front-end to listen to the dedicated broadcast channel (i.e., Rx mode). Once the broadcast is received, the controller extracts the scan-list and instructs the RF-front end to measure the RSSI values. Similar to the wired sensors, the controller fetches the RSSI values and switches the RF front-end in the Tx mode and transmits the data to the hub. Once done, it switches back to the Rx mode and the process repeats.

5.3.3.3 Hubs

The role of a hub is not only to communicate with the wireless sensors in a "hub and spoke" configuration but also to serve as a wired sensor. The hub is equipped with the RF-frond end, controller and Ethernet interface. While sensing, it simply behaves just like the wired sensors.

While controlling the wireless sensors, the hub is responsible for i) broadcast of scan list to respective sensors (i.e., task allocation) ii) uplink channel allocation to individual wireless sensors, and iii) data aggregation and UDP streaming.

On power-on, the controller contacts the DSA server and obtains a scan list for itself and the wireless sensors connected to it. As in the wired sensors, the controller cycles through the list of frequencies, configures the RF front-end, fetches the data and streams it over UDP to the DSA server. Next, it configures its RF front-end for transmission on the broadcast channel and transmits the scan-list to the wireless sensors. It can be noted that the hub might instruct *all* the wireless sensors to scan the same set of frequencies or it might instruct the sensors to scan different sets of frequencies.

Once done, the hub switches the RF front-end to the receive mode and waits for the wireless sensors to send their data. Transmissions by the wireless sensors are scheduled by the hub using any mechanism such as polling or token passing. Depending on the time-criticality of the RSSI measurements the hub can i) have a single RSSI measurement per packet, ii) aggregate multiple RSSI measurements from the same wireless sensor into a UDP packet, or iii) aggregate multiple RSSI measurements from different wireless sensors into a UDP packet. In all three cases, it streams the UDP packet to the DSA server using its Ethernet interface.

5.4 DSA server

As discussed in the system architecture, the DSA server has four primary components: RSSI collector, real-time diagnostics, configuration panel, and database engine.

5.4.1 RSSI Collector

The RSSI collector can be a single or multi-threaded server depending on the number and size of RSSI incoming UDP streams from the sensors and hubs. The RSSI collector extracts all the fields (i.e., nodeID, location, freq, bandwidth, and RSSI values) from the UDP packets and time-stamps and passes them to the database engine. The same information is also passed to the real-time diagnostics module. Apart from the RSSI streams, the RSSI collector also gets the queries for the scan-list from the sensors and hubs which is directed to the configuration panel.

5.4.2 Real-time Diagnostics

Diagnostics allow spectrum managers to have real-time assessment of the spectrum assay of the RF field being sensed. Various representation techniques and/or display visualization can be used that best conveys the meaning of the parameters captured by the sensing network.

Based on the assessment, it also provides a feedback to the configuration panel for scan list parameters update.

5.4.3 Configuration Panel

The configuration panel is used to configure the scan parameters for the sensors and hubs. Such configuration can be done in an automated manner in response to what the diagnostics are or it could be done manually by the end-user. In either case, these parameters (target frequencies, bandwidth, sweep time, and scan resolution) are sent to the sensors and hubs.

5.4.4 Database Engine

This is a multi-threaded server which receives the time-stamped entries from the RSSI collector. This engine provides a mechanism for storage and retrieval of data by storing data in a `NoSQL` format, primarily because of its simplicity and scalability. If needed, the engine purges entries which are older than a specifies time period. It also replies to all the queries from the service engine.

5.5 Service Engine

A variety of services can be provided to the end users, network operators, and spectrum managers based on the available information at the DSA server. Such services are facilitated by the service engine which could be either: i) collocated with the DSA server, or ii) cloud-based. To provide ease of implementation and provide modularity to the wide variety of services, the service engine uses two important modules as shown in Fig. 5.1. The service mapper is a multi-threaded TCP and UDP server which takes in the user inputs with varying formats and directs them to the right service. The database interface provides a single API for the various services to request database entries.

The services themselves are some methods that make use of the database entries and supplies meaningful information to the requester which could be an end-user or a DSA node. Fig. 5.1 shows three different services: i) *channel allocation*, ii) *radio environment map*, and iii) *real time spectrum monitoring* (these 3 services have also been implemented as discussed in Section 5.6).

5.5.1 Channel Allocation

For any channel allocation and access, it is important for a pair of transceivers to know the channel that provides the best possible data rate.

This is even more crucial in a DSA system because of the fleeting nature of the set of available channels. The channel allocation service finds the best available channel for the given locations of the transmitter and the receiver. As the database might not contain the RSSI values for requested locations, channel allocation service resorts to some interpolation/extrapolation methods to estimate the data rate for the given locations.

5.5.2 Radio Environment Map

A REM is a representation of the RF field in space that is obtained by sampling the field by the spectrum sensors at various locations. It typically represents the power spectral density for the target frequency for a given region. The number of points for which the power spectral density has to be estimated depends on the temporal and spatial granularity of the desired REM, thus a more detailed REM requires more computations.

5.5.3 Real-time Spectrum Monitoring

This service allows the end user to monitor the RSSI measurements by any subset of the spectrum sensors in real-time. This is achieved by streaming the RSSI values over the Internet to any host. It also offers a variety of ways to graphically display the data in a user friendly manner.

5.6 Hardware and Software Implementation

Based on the proposed DSA system architecture and capitalizing on the SCRF chips' capabilities, we implement a low-cost system level solution for the same. For the sensing network, we design and build four different kinds of sensors– three of which are wired and one is wireless. The DSA server was implemented on a Linux machine. As for the services, we implement three services: channel allocation to the DSA enabled nodes, creation of the radio environment map, and real-time spectrum monitoring. Let us discuss each of the components in detail.

5.6.1 Wired Sensors

We built three types of wired sensors that had different sensing and streaming capabilities. For all three, *to ensure compatibility*, we used the same frame format to stream the UDP packets to the DSA server:

```
nodeId:targetFreq:BW:RSSI:XX:YY
```

The $nodeId$ is used to identify the sensor, the $targetFreq$ and BW are the center frequency and bandwidth being scanned, $RSSI$ is the measured signal strength in dBm, XX and YY are the X-coordinate and Y-coordinate of the sensor. The operational procedure for the wired sensor is shown in Fig. 5.4. Next, we discuss the The 3 types of wired sensor are:

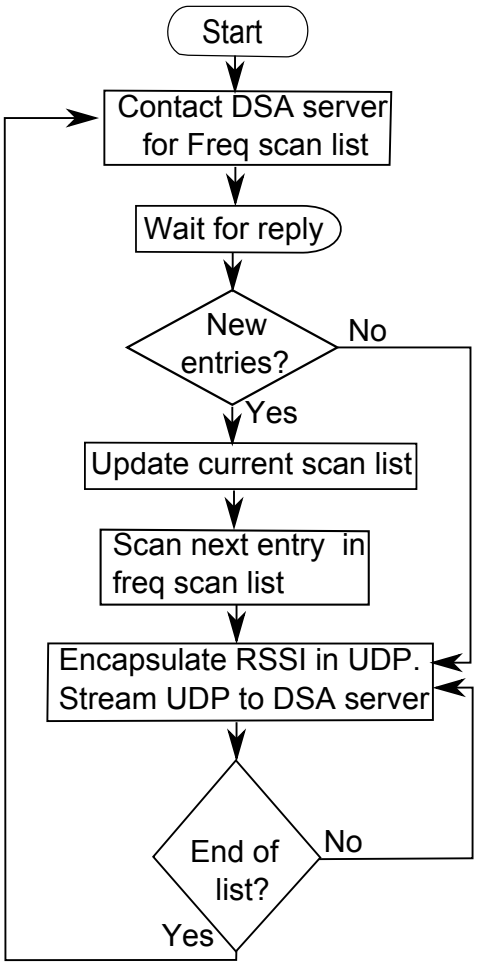


Figure 5.4 Operation procedure for the wired sensor.

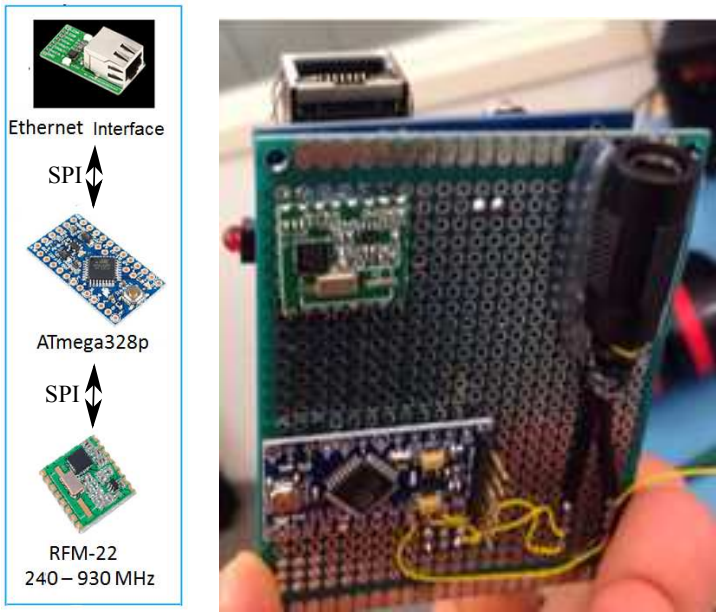
Type-1 Sensor: Type-1 sensor is meant to detect sub-GHz frequencies. All the components were low-cost off-the-shelf devices– the total cost of which was less than $10. The individual components are shown in Fig. 5.5(a) while the assembled one is shown in Fig. 5.5(b).

RF Front-end: The RF front-end for this sensor is RFM22 by HopeRF [106] which has a frequency response from 240 MHz to 930 MHz. The receiver sensitivity is -118 dBm with a configurable receiver bandwidth from 2.6 KHz to 620 KHz. It also has an RSSI reporting feature. The *transmit* capabilities of RFM22 are exploited for the hub and the wireless sensors as will be discussed in sections 5.6.2 and 5.6.3.

Controller: We used the microcontroller ATmega328p as the controller. It is an 8 bit device with 32 KBytes of ROM, 2 KBytes of RAM, with 20 MIPS. Packet encapsulation and streaming were done as follows.

```
String DSAServerIP=10.192.168.4;
define RF-SPI-PIN 5
define Ethernet-SPI-PIN 4
scanList []= ObtainScanList();
For(each entry i in scanList)
      // shared SPI requires individual activation of RF and Eth.
      SwitchSPIControl(5);
      setFreq(scanList[i]);
```

```
        double reading=readRssi();
        double RSSI=convertReadingToRSSI(reading);
        String payLoad = buildPayLoad(nodeId, scanList[i],
                              BW, RSSI, XX, YY);
        SwitchSPIControl(4);
        delayMicroSecond(10);
        selectEthernetInterface();
        Datagram udp=Datagram(DSAServerIP, payLoad);
        sendUDP(udp);
End-For-i
```



(a) (b)

Figure 5.5 (a) Components for Type-1 sensor; (b) Assembled Type-1 sensor.

Ethernet Interface: We used W5100 chip which is a hardwired TCP/IP embedded Ethernet controller that enables easier Internet connection for embedded systems. An attractive feature of this chip is that it can be controlled via SPI by the ATmega328p microcontroller.

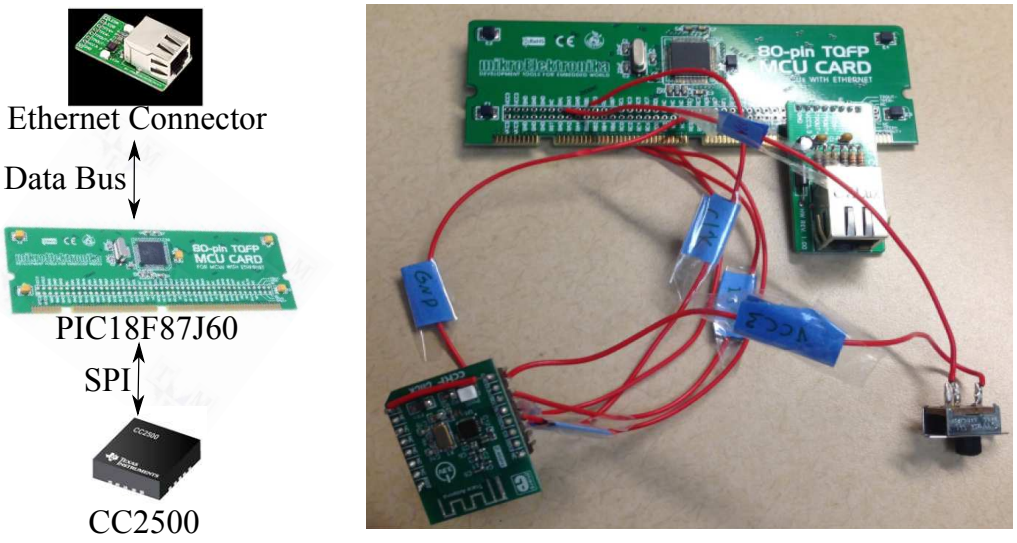
Type-2 Sensor: Type-2 sensor is meant to detect the 2.4 GHz ISM band. Again, all components were off-the-shelf with the total cost being less than $10. The components are shown in Fig. 5.6(a) while the assembled sensor is shown in Fig. 5.6(b).

RF Front-End: The RF front-end of this sensor uses CC2500 by TI which has a frequency response from 2.4 GHz to 2.4835 GHz [109]. The receiver sensitivity is -104 dBm with a configurable receiver bandwidth from 58 KHz to 812 KHz. CC2500 also has an RSSI reporting feature. Though CC2500 has transmit capabilities, they are not needed for RSSI sensing.

Controller and Ethernet Interface: We used the microncontroller PIC18F87J60 which is an 8 bit device with 128 KBytes of ROM, 3808 Bytes of RAM, and processes up-to 10 MIPS. This microcontroller has a built-in 802.3 compatible Ethernet controller with integrated PHY and MAC modules.

```
String DSAServerIP=10.192.168.4;
scanList []= ObtainScanList();
For(each entry i in scanList)
        setFreq(scanList[i]);
        double RSSI=readRssi();
        String payLoad = buildPayLoad(nodeId, scanList[i],
```

```
                BW, RSSI, XX, YY);
        Datagram udp=Datagram(DSAServerIP, payLoad);
        sendUDP(udp);
End-For-i
```



(a) (b)

Figure 5.6 (a) Components for Type-2 sensor; (b) Assembled Type-2 sensor.

Type-3 Sensor: Type-3 sensor is meant to detect a wider band (50 MHz to 1.87 GHz); however, it is costlier than the other two with a total cost of $45. The assembled sensor along with the internals of the USB dongle is shown in Fig. 5.7.

RF Front-End: We used a USB dongle based TV tuner and reverse-engineered it to serve as the RF front-end. The dongle contains two essential RF chips: the first is the RF tuner chip R820 and the second is the demodulator chip RTL2838. The combination of these two chips allows reconfigurable frequency from 50 MHz to 1.87 GHz with a configurable bandwidth of up to 2.4 MHz while supporting ASK, FSK PSK and QPSK demodulation

schemes. The receiver's sensitivity is between -72 dBm and -92 dBm and depends on the bandwidth and demodulation scheme used.

It is to be noted that this dongle is sold for decoding 6 MHz TV channels. It does not directly measure RSSI values nor does it allow any 'receive bandwidth' that is less than or more than 6 MHz. In order to measure RSSI for any bandwidth, we implemented customized device drivers to communicate with the PIC micro-controller inside the dongle to pass the modified commands to the R820T tuner and the RTL2838 demodulator chip to obtain the raw demodulated data. Fast Fourier Transform (FFT) is performed on the raw data with the desired resolution (i.e., 100 pins per 1 MHz) that yields the RSSI values.

Controller and Ethernet Interface: We used BeagleBone which is a single board computer (SBC) due to the following reasons: i) the USB dongle requires `libusb-1.0` support for USB interface, ii) capability to handle large amounts of raw demodulated data for FFT computation, iii) complete USB host functionality to execute the customized device drivers, and iv) integrated Ethernet support.

```
String DSAServerIP=10.192.168.4;
 int pinSize=100;
scanList []= ObtainScanList();
For(each entry i in scanList)
       setFreq(scanList[i]);
       byte [] baseBand=captureData();
       double [] readings=FFT(baseBand);
```

```
double RSSI=readingsToRSSI(reading);
String payLoad = buildPayLoad(nodeId, scanList[i],
                              BW, RSSI, XX, YY);
Datagram udp=Datagram(DSAServerIP, payLoad);
sendUDP(udp);
End-For-i
```

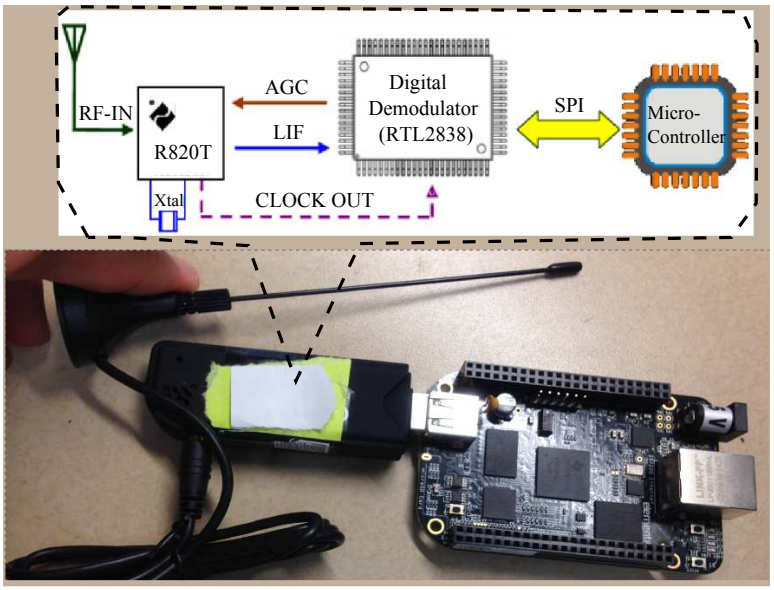


Figure 5.7 Type-3 sensor.

A sample wide-band scan, 50 MHz to 1.7 GHz, performed with a Type-3 sensor is shown in Fig. 5.8. The scan was performed on December 12th 2015 around 04:46 PM eastern time at Lat: 28.6035429, Long: -81.1994791. The scanning resolution was set to 100 pin/MHz.

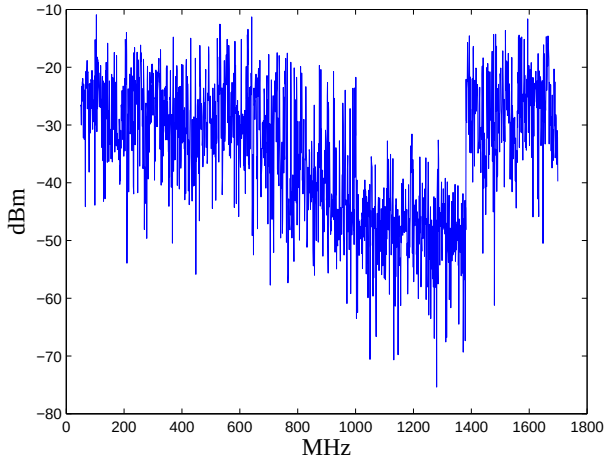


Figure 5.8 Measurements performed by Type-3 sensor.

5.6.2 Wireless Sensors

The wireless sensors have the same RF front-end and the controller as that of Type-1 sensors. Being wireless, it does not have Ethernet connectivity. For transmission, it uses the same RF front-end that is used for sensing. For the Tx part, the RFM22 supports FSK, GFSK, and OOK modulation schemes with data rates from 1 to 128 Kbps. It has configurable transmit power from 8 dBm up to 17 dBm. The operational cycle of the wireless sensor is shown in Fig. 5.9.

The wireless sensors send data to the hub using a customized light-weight protocol using the following format:

```
MyNodeID:targetFreq:BW:RSSI:XX:YY
```

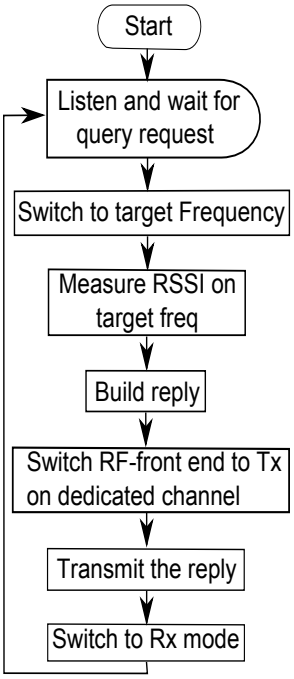


Figure 5.9 Operation procedure for wireless sensor.

The `MyNodeID` is the id of the sensor that measured the RSSI, XX and YY are the x and y coordinates of the sensor, $targetFreq$ is center frequency of the band being sensed, BW is the bandwidth, and RSSI is the measured value. The components of a wireless node are shown in Fig. 5.10(a) while an assembled wireless sensor is shown in Fig. 5.10(b).

5.6.3 Hub

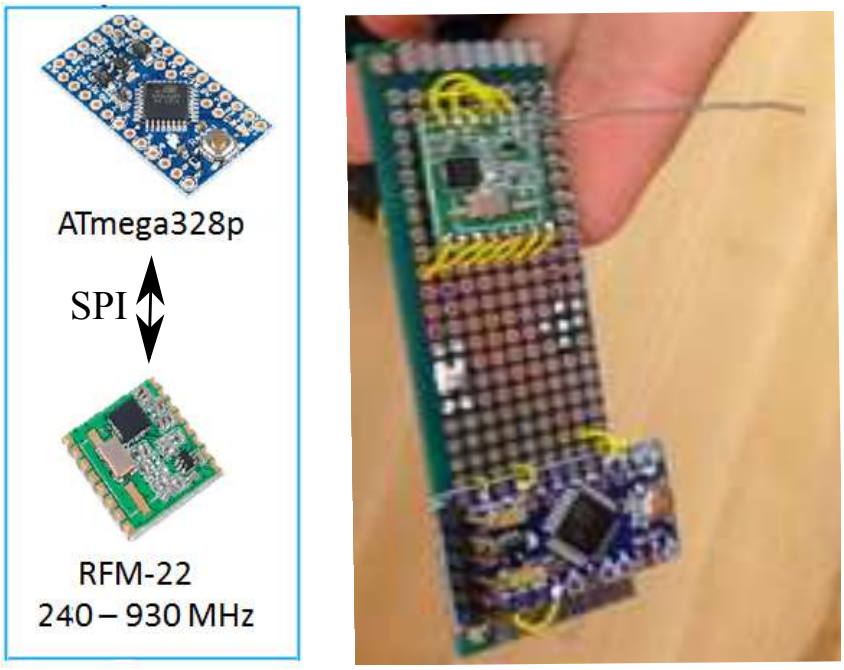
We used the same hardware components that we used for Type-1 sensor to serve as the hub for the wireless nodes. In addition to the functions that are performed by the Type-1 sensor, the hub has been programmed to query the wireless nodes, manages their channel access

using a polling-based MAC, get their responses, encapsulate the data in UDP packets, and stream them to the DSA server. The operational cycle is show in Fig. 5.11. An assembled hub is identical to the wired sensor shown in Fig. 5.5(b).

Using a dedicated channel, the hub queries the wireless nodes in its vicinity using the frame format:

`YourNodeID:targetFreq:BW`

where The `YourNodeID` is the id of the sensor that is being queried by the hub, targetFreq is center frequency of the band to be sensed, and BW is the bandwidth. Once the hub receives the data from a wireless sensor, it encapsulates the entire data in a UDP packet and streams it to the DSA server.



(a) (b)

Figure 5.10 (a) Components for wireless sensor; (b) Assembled wireless sensor.

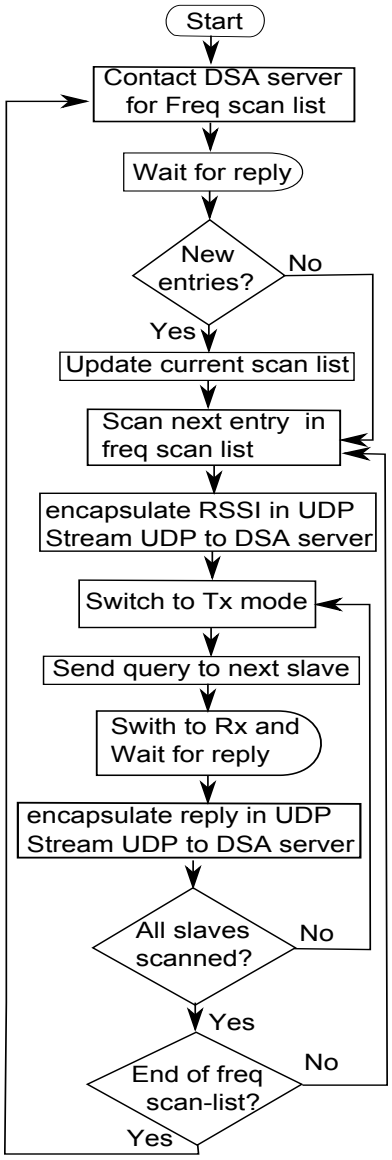


Figure 5.11 Operation procedure of the hub.

Figure 5.12 Measurements performed at the hub (unit A) and the wireless sensor (unit B).

Sample wide-band scans, 400 MHz to 930 MHz, performed by a hub and a wireless sensor are shown in Fig. 5.12. The scans were performed at January 28th 2016 around 09:46 PM eastern time at Lat: 28.6035429, Long: -81.1994791.

5.6.4 DSA Server

The DSA server, consisting of the four modules, is the core of the entire system. We implemented all the modules in Jave on a Linux box running Ubuntu 14.04.

For the RSSI collector, a multi-threaded server is run on port 25100 (randomly chosen) to read the incoming UDP packets. To infer the presence of primaries, we implemented the classical periodogram in Java that essentially compares the RSSI values against a threshold.

The diagnostic functionality keeps track of the active primaries and their spectrum usage patterns. A multi-threaded UDP server was implemented to allow the sensors in the field to open a UDP connection to the configuration panel. This connection was used by the sensors to request the scan-list from the configuration panel and also for the configuration panel to respond with the same. The UDP frame format that was used was:

$$nodeID : freq_1 : BW_1 : freq_2 : BW_2 : ... : freq_N : BW_N$$

with the $nodeID$ being the id of the node asking for the scan-list, BW_i is the bandwidth for the i-th channel around the target center frequency $freq_i$. Using the configuration panel, Type-1 sensors were configured to scan channels with bandwidth of 300 KHz centered at 900, 905 and 910 MHz. Type-2 sensors were configured to scan frequencies from 50 MHz to 1.7 Ghz with channel width of 2 MHz. Type-3 sensors were configured to scan from 2.4 GHz to 2.485 with a bandwidth of 50 KHz. The RSSI reported value, nodeID, node's location, bandwidth, target frequency from all sensors were time-stamped and stored as 'Comma Separated Values' (CSV) file format in the database engine as:

```
a,b,c,....
```

5.6.5 Service Engine

We implemented three services as part of the service engine. All services made use of the Database Interface and the Service Mapper. The database interface was implemented as a Java-API that takes a query from the services and parses the database entries and returns the results to the respective service. The service mapper was implemented as a multi-threaded UDP server on the following ports 25200, 25300 and 25400 corresponding to the channel allocation, radio environment map, and real-time spectrum monitoring services respectively.

Channel allocation service: was implemented as a multi-threaded Java server running on port 25200. This service is used by the DSA enabled transmitter-receiver pairs as they seek the best vacant channel in their vicinity. Upon receiving a channel allocation request with the format

```
nodeId:XX:YY
```

the channel allocation service extracts the pairs' location (i.e., (x, y) coordinates). It then accesses the database engine to find entries that might have been recorded by sensors in the viciniy of (x, y). We used a modified version of Shepard's interpolation technique to estimate the RSSI (and hence the best possible channel) for the location in question by fusing the RSSI values recorded by sensors in the vicinity [110]. In particular, we implemented the technique proposed in [111] which was computationally lightweight and also efficient for different types of primary networks. The service responds back with a UDP packet using

the following format:

`nodeId:freq:BW`

where `freq, BW` are the frequency and bandwidth of the recommended channel.

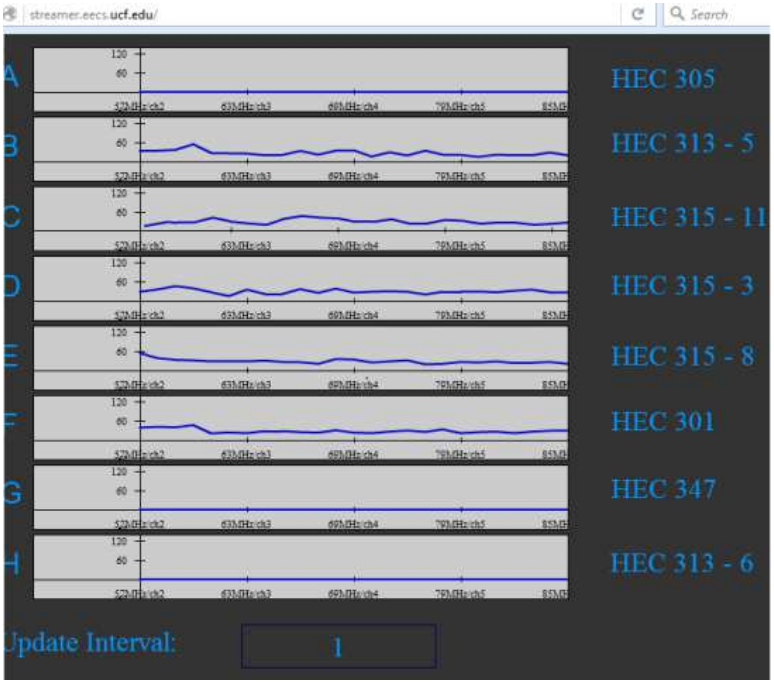


Figure 5.13 Client side of real time monitoring service showing 5 live strems.

Real-time monitoring service: was implemented as a multi-threaded TCP server on port 25300. Upon receiving a TCP request from the user/client, the service uses the database interface to obtain the most recent RSSI values from all the database engine and streams them to the user using the frame format:

$$nodeID : freq_1 : BW_1 : freq_2 : BW_2 : ... : freq_N : BW_N$$

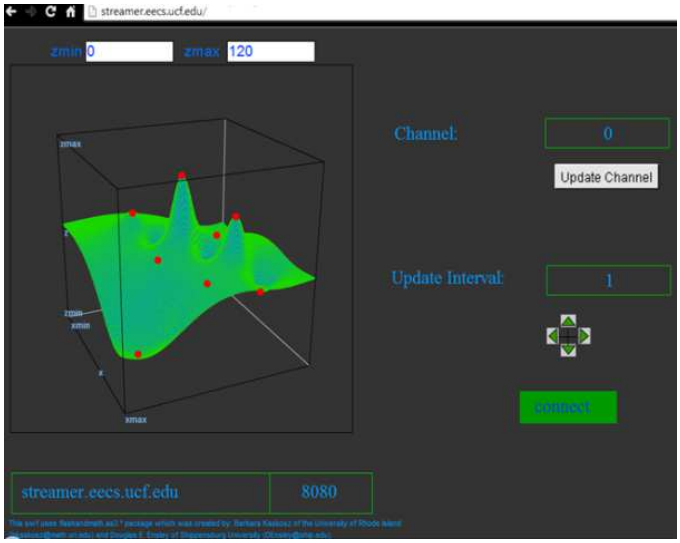


Figure 5.14 Radio environment map displayed as 3D map at the client side.

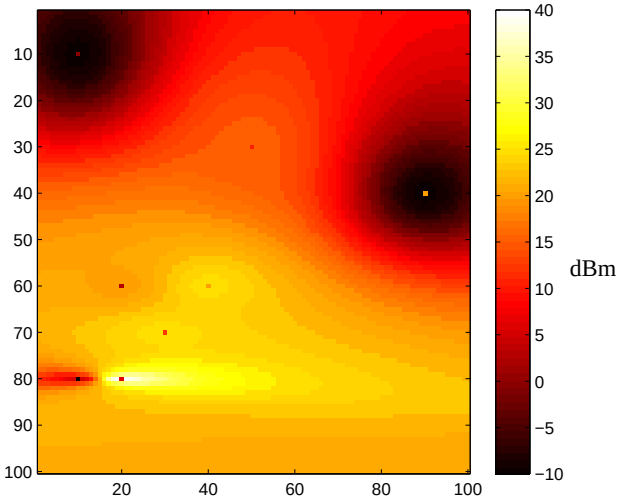


Figure 5.15 Radio environment map as displayed as heat map at the client side.

Fig. 5.13 shows the real time monitoring service at the client side.

Radio environment map: is similar to the channel allocation service; however, instead of estimating for one location, the REM is constructed for $N \times N$ discrete locations in the area of interest. One time instance of the map is computed as:

```
define mapResolution 1
define mapWidth 100
define mapLength 100
int dataPoints=mapWidth/mapResolution;
// location[0] is X, location[1] is Y
double location[]=extractLocationFromRequest();
 double yUpperLimit= location[1]+mapHeight/2;
double yLowerLimit= location[1]-mapHeight/2;
double xUpperLimit= location[0]+mapWidth/2;
double xLowerLimit= locaction[0]-mapWidth/2;
double  sensorReadings[]= QuerydataBaseforSensorsWithin(
                          xUpperLimit , xLowerLimit, yUpperLimit,
                          yLowerLimit);
For (xIndex: xIndex<mapWidth: stepSize=mapResolution)
        For (yIndex: yIndex < mapLenth: stepSize = mapResolution)
                estimatedRSSI = modifiedShepard(sensorReadings, xIndex,
                                                yIndex);
                payload=concat(payload, xIndex, yIndex, estimatedRSSI);
```

```
        End-For-y
EndFor-For-x
        Datagram udp=Datagram(requesterIP, payload);
        sendUDP(udp);
```

The results are streamed to the user using following frame format:

$$X_1 : Y_1 : RSSI_1 : X_2 : Y_2 : RSSI_2 : \cdots : X_N : Y_N : RSSI_N$$

where $RSSI_i$ is the estimated RSSI value at coordinate X_i and Y_i.

Figs. 5.14 and 5.15 show two different representation of the radio environment map as displayed at the client side.

5.7 Summary

In this chapter, we presented and implemented a low-cost yet effective architecture that enables dynamic spectrum access for any network ranging from IoT to cellular. We broke the cost-scalability barrier and showed that a complete system level solution for a database-assisted DSA system can be implemented with standard servers and inexpensive software configurable RF chips, thereby achieving economics of scale.

First, we presented the overall architecture and the system components. Next we designed each component and showed how it integrates within our architecture. For the RSSI

sensing network, we built wired and wireless spectrum sensors that operate on 280–930 MHz, 50 MHz–1.87 GHz, and 2.4–2.5 GHz using low-cost off the shelf software configurable RF chips. To get the RSSI values on a set of bands, we use generic micro-controllers to program the operating parameters (scan range, center frequency, bandwidth resolution, demodulation scheme and scan rate) of the SCRF chips. The wireless sensors transmit the sensed RSSI values to the nearest Ethernet-enabled hub using a light-weight communication protocol. The hub aggregates the data from multiple sensors and streams to the DSA server using UDP over IP. On receiving the real-time RSSI values from various sensors, the DSA server stores them in database engine with other meta data. This database is made accessible by the service engine that can offer a variety of services. We implemented three services: channel allocation, radio map creation, and real-time spectrum usage monitoring.

CHAPTER 6: SIMULATION MODEL AND RESULTS

In this chapter, we discuss the simulation models, experiments and testbeds. We first present the results on connectivity, followed by the results on capacity. Finally we do a performance evaluation for the implemented DSA system.

6.1 Connectivity

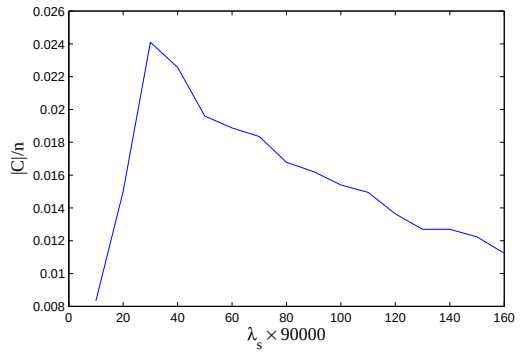
6.1.1 Connectivity of Interference Limited DSA Networks

To validate the theoretical findings on connectivity of DSA networks, we conduct Linux based simulation experiments. In particular, we seek to find thinning probability, percolation conditions, coverage shrinkage, and the effective density. We consider two different Poisson distributed secondary networks thus eliminating any bias towards a favorable network setting. The first network is discussed in 6.1.1.1 and the second network is discussed in 6.1.1.2. We use $|C|$ to denote the size of the biggest component and n to denote the total number of deployed secondary nodes. The ratio $|C|/n$ denotes the relative size of the biggest connected component. We use the word percolation to refer to the formation of a spanning giant component which contains at least 40% of the deployed nodes.

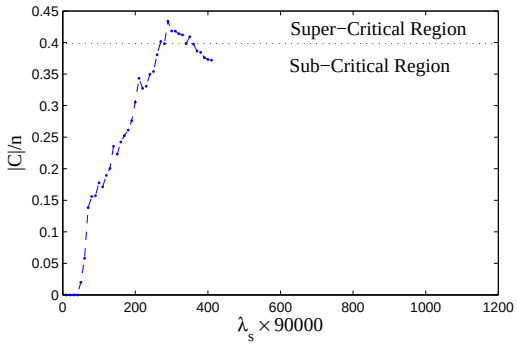
6.1.1.1 Network-1

We consider an area of 300×300 with secondary density of λ_s, $r_s = 25$, $r_I = 35$, $\alpha = 2$, and $\beta = 1$. To illustrate Lemma 3.1, we set $\gamma = 0.5$ and $\lambda_p = 0$. The resultant connectivity is shown in Fig. 6.1(a). Using the Boolean model, it was found that a network with $r_s = 25$ will percolate once the deployment density is more than 151/90000 i.e., $\lambda_s^c = 151/90000$. In other words, more than 151/90000 nodes per unit area should be visible in order for the percolation to occur. By examining Fig. 6.1(a) it can be seen that the network never percolates although λ_s is driven greater than λ_s^c. The reason can be explained by checking Fig. 3.8. It can be seen that, P_{thin} is very high for $\lambda_s > 151/90000$ and approaches 1. Thus the percolation visible nodes have density of $\lambda_s \times (1 - 0.9999)$ which is less than 151/90000 and that is why percolation can never occur for such a network.

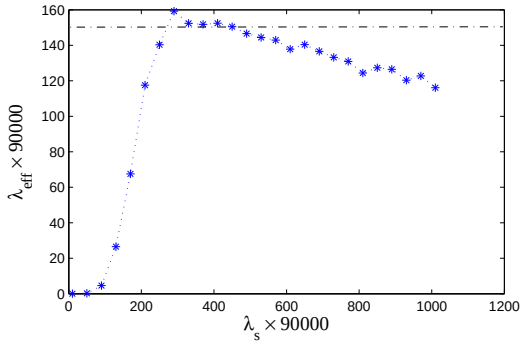
Next, we keep $\lambda_p = 0$ and set $\gamma = 0.08$ and observe the resultant connectivity as shown in Fig. 6.1(b). It can be noted that the network percolates for a certain range of λ_s values while it does not for the rest. To understand such behavior, we examine the analytical plot of λ_{eff} for $\gamma = 0.08$ as shown in Fig. 6.1(c) and observe that although λ_s goes from 0 to 1200/90000 only a portion of it produces $\lambda_{eff} > 151/90000$ at which percolation occurs. For further understanding, we revert to the analytical plots of P_{thin} (Fig. 6.1(d)), P_{d-x} (Fig. 6.1(e)), p_{dia}, p_{side} (Fig. 6.1(f)), and the reduced coverage radius (Fig. 6.2(a)) for $\gamma = 0.08$. From Fig. 6.1(e), we note that at low densities, P_{d-0}, p_{side}, and p_{dia} dominate, due to deployment scarcity. As λ_s increases, nodes start to have a neighbor in their r_s, this



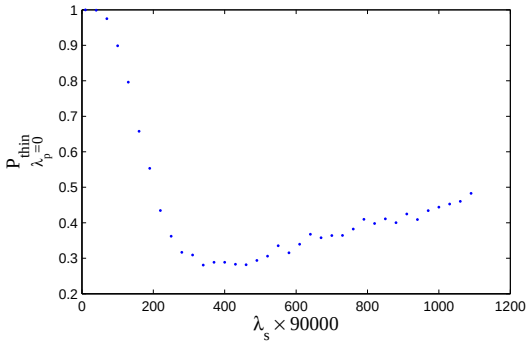
(a)



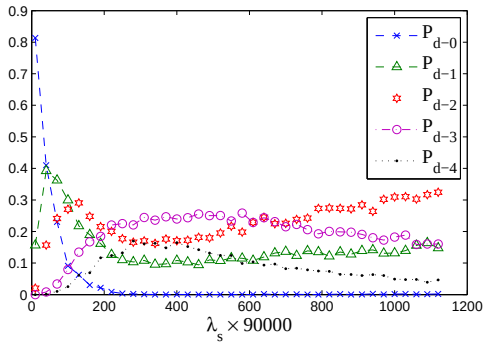
(b)



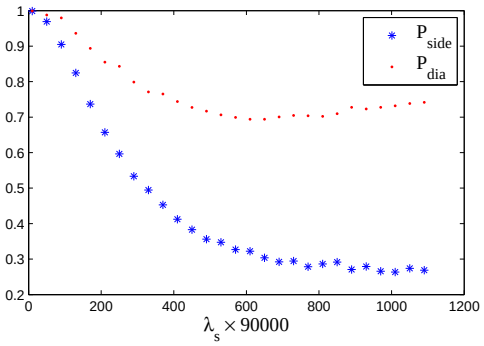
(c)



(d)



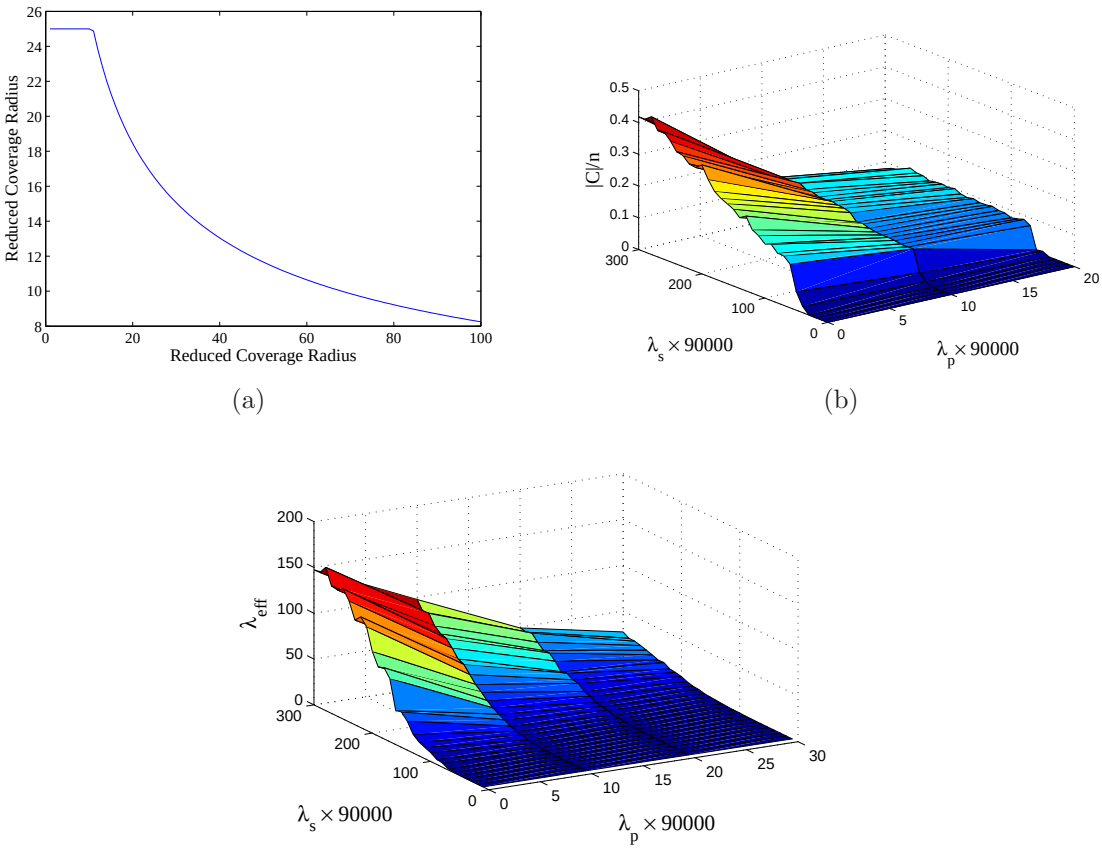
(e)



(f)

Figure 6.1 A) $|C|/n$ versus λ_s at $\gamma = 0.5$; B) $|C|/n$ versus λ_s at $\gamma = 0.08$; C)Effective density versus λ_s with $\gamma = 0.08$, $\lambda_p = 0$; D) Total thinning probability versus λ_s with $\gamma = 0.08$, $\lambda_p = 0$; E) Theoretical plot for $P_{d-0}, \cdots P_{d-4}$ versus λ_s at $\gamma = 0.08$; F)Theoretical plot for P_{side} and P_{dia} versus λ_s at $\gamma = 0.08$.

is why P_{d-1} and P_{d-2} peak. Then, the number of neighbors increases which gives a rise to P_{d-3} and P_{d-4} while P_{d-0}, P_{d-1}, and P_{d-2}, p_{side} and p_{dia} decrease. This leads to decrease in P_{thin} and increase in λ_{eff} to more than 151/90000 at which percolation occurs. On deeper examination, we note that, $|C|/n$, λ_{eff}, P_{d-3}, and P_{d-4} maximize for the same values of λ_s which minimize P_{thin}, P_{d-0}, P_{d-1}, P_{d-2}, p_{side} and p_{dia}. For large values of λ_s, the distance between the nodes becomes small. For this reason, the nodes start to dominate on each other increasing P_{d-1} and P_{d-2} again.



(a) (b)

(c)

Figure 6.2 A) Shrinkage in coverage radius versus number of interferes for $\gamma = 0.08$; B) Analytical plot for $\lambda_e f f$ versus λ_s and λ_p with $\gamma = 0.08$; C) Analytical plot for $\lambda_e f f$ versus λ_s and λ_p with $\gamma = 0.08$.

This explains the behavior of λ_{eff} in Fig. 6.1(c) where it increases then decreases although λ_s keeps increasing from 0 to 1200/90000 nodes/unit-area, however only a portion of λ_s values[1] results in $\lambda_{eff} > 151/90000$ and that is where percolation occurs in the simulated network. We also note that as the number of percolation visible nodes per unit area (i.e., λ_{eff}) increases so does the corresponding connectivity.

To illustrate the effects of the primary users on the current simulated secondary network, we simulate a primary network with density $r_p = 35$ and a set of values for λ_p. We show the resultant connectivity in Fig. 6.2(b). The corresponding analytical plot of λ_{eff} along λ_s-λ_p values is shown in Fig. 6.2(c). As expected, presence of primary users degrades the secondary connectivity; while the secondary network percolates only for the λ_s-λ_p values which result in $\lambda_{eff} > 151/90000$.

6.1.1.2 Network-2

For the second network, we consider a deployment area of 500×500 with $r_s = 55$, $r_I = 65$ and $\gamma = 0.06$. At first, λ_p is set to 0 (i.e., overlay) to emphasize the secondary-to-secondary interference. Eqns. (3.2), (3.6), (3.9), (3.10) and (3.14) are used to find $\bar{r}_s$, $P_{d=M}$, P_{inter}, P_{side}, and P_{iso} respectively. These are then used in Eqn. (3.5) to obtain λ_{eff}. The corresponding analytical plots are shown in Figs. 6.4(a), 6.4(b), 6.4(c), 6.4(d), and 6.3(a), while the resultant connectivity is shown in Fig. 6.3(b).

[1]Those are the λ_s values that correspond to λ_{eff} values which are above the dotted line

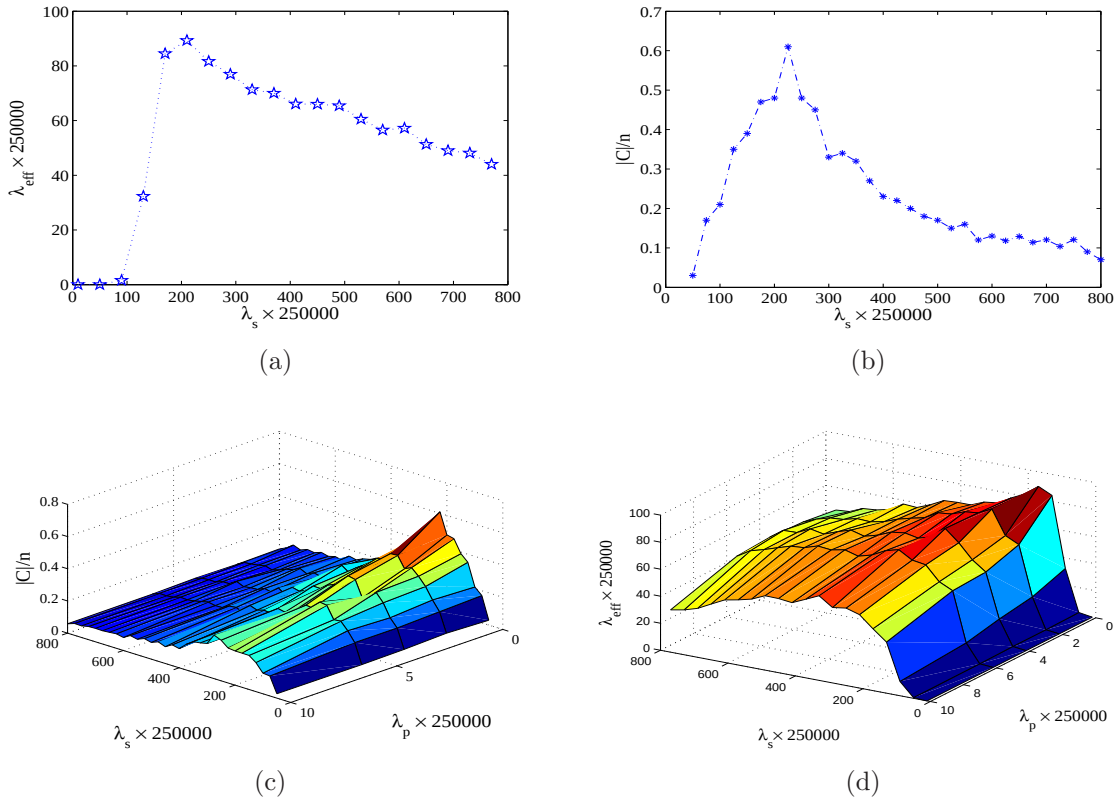


Figure 6.3 A) λ_{eff} versus λ_s for $\gamma = 0.06$; B) $|C|/n$ versus λ_s for $\gamma = 0.06$; C) $|C|/n$ versus λ_s and λ_p for $\gamma = 0.06$; D) Analytical plot for λ_{eff} vs λ_s and λ_p for $\gamma = 0.06$.

In this network as well, it can be noted that, although λ_s goes from 0 up to 800/250000, only a portion of it results in $\lambda_{eff} > 70/250000$ (Fig. 6.3(a)) at which percolation occurs as illustrated in Fig. 6.3(b). We point out that, 70/250000 is λ_s^c for the Boolean network with $r_s = 65$. We also notice that, an increase in λ_{eff} results in an increase in the resultant connectivity as illustrated in Fig. 6.1(b), and 6.1(c) for network-1 ($\gamma = 0.08$) and Figs. 6.3(a) and 6.3(b) for network-2.

To show the effects of the primary users, we simulate a primary network with $r_p = 55$ and a set of values for λ_p. The resultant connectivity is shown in Fig. 6.3(c). The corresponding analytical plot of λ_{eff} along λ_s-λ_p values is shown in Fig. 6.3(d). Again, it

can be noted that, presence of the primary users degrades the connectivity; while percolation occurs for λ_s-λ_p values which result in $\lambda_{eff} > 70/250000$.

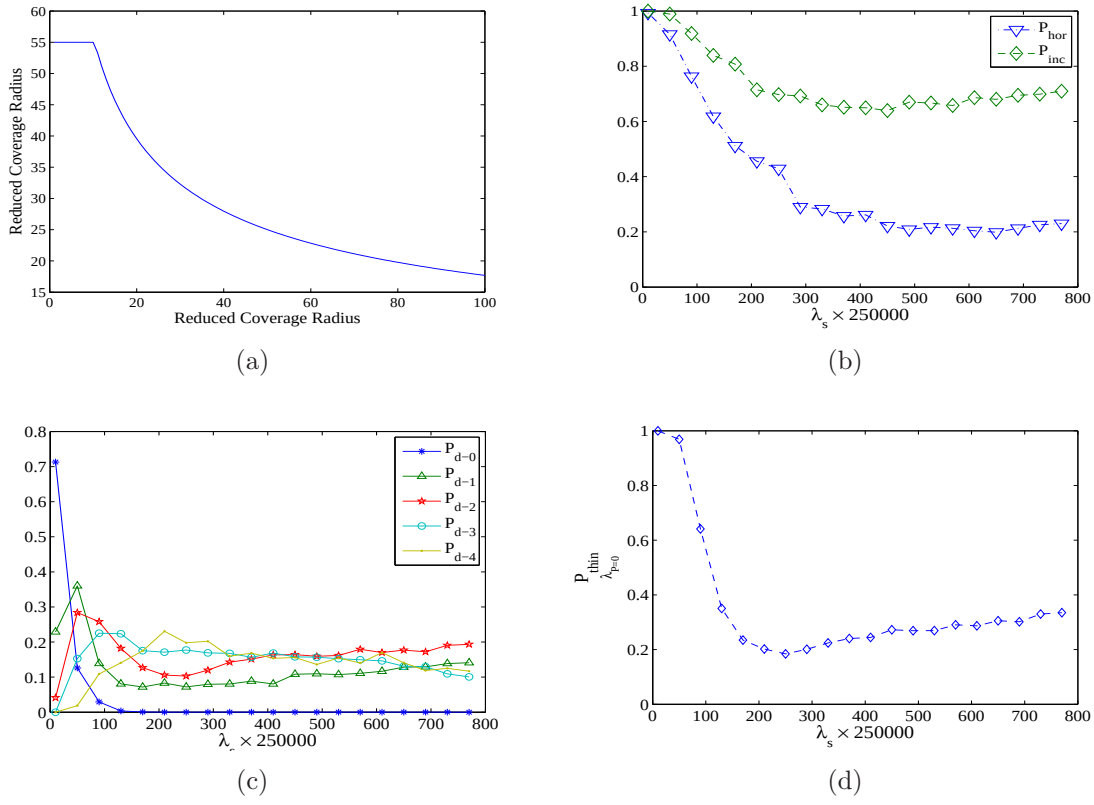


Figure 6.4 A) Shrinkage in coverage radius versus number of interferes for $\gamma = 0.06$; B) Theoretical plot for P_{side} and P_{dia} for $\gamma = 0.06$; C) Theoretical plot for $P_{d-0}, \cdots P_{d-4}$ versus λ_s for $\gamma = 0.06$; D) Total thinning probability versus λ_s for $\gamma = 0.06$;

6.1.2 Connectivity Maximization

For network-1 and network-2, we show the corresponding results of their optimal density, optimal receive ratio and optimal TDMA slotting. We use $\gamma = 0.08$ for network-1 and $\gamma = 0.06$ for network-2.

6.1.2.1 Optimal Density

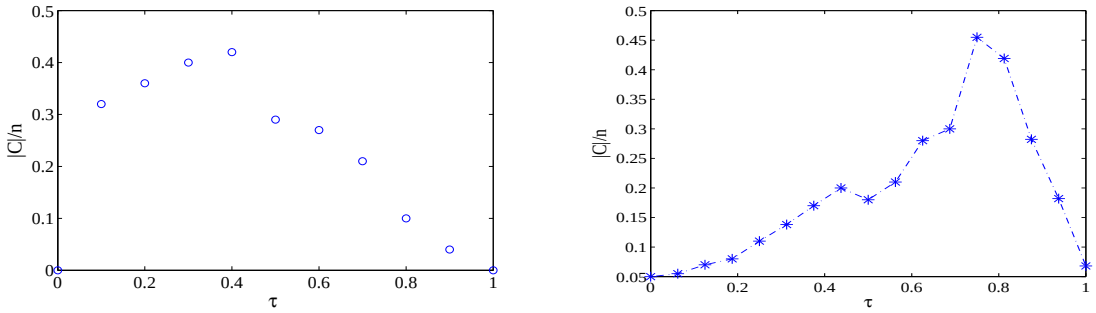
The goal here is to compare λ_s^{opt} values for both networks which are obtained analytically and via simulations. For network-1, solving Eqn. (3.20) numerically results in $\lambda_s^{opt} = 300/90000$. Simulation results (Fig. 6.1(b)) show that the maximum connectivity is attained at $\lambda_s^{opt} = 315/90000$. Solving Eqn. (3.20) for network-2 results in $\lambda_{opt} = 217/250000$. The simulation results (Fig. 6.3(b)) shows that the maximum connectivity is attained at $\lambda_s^{opt} = 225/250000$.

6.1.2.2 Optimal Receive-only Ratio

The goal here is to compare τ^{opt} for both networks obtained analytically and via simulations. For network-1 under $\lambda_s = 500/90000$, Fig. 6.1(b) shows that the corresponding connectivity is around 0.35 and the network is not percolated. To maximize the network's connectivity, we use the random selection approach. Substituting $\lambda_s^{opt} = 300/90000$ (from previous section) in Eqn. (3.21) results in $\tau^{opt} = 200/500 = 0.4$. To verify the analytical value of τ^{opt}, we simulate the network with $\tau = [0, 1]$ and show the resulting connectivity in Fig. 6.5(a). The simulation result shows that the highest connectivity is attained at $\tau = 0.4$.

For network-2, under $\lambda_s = 800/250000$, Fig. 6.3(b) shows that the resulting connectivity is around 0.09 where the network is not percolated. Again, we use the random selection approach to increase the connectivity. Using the analytical value that was found for $\lambda_s^{opt} = 217/250000$ in Eqn. (3.21) results in $\tau^{opt} = (800 - 217)/800 = 0.72875$. To ver-

ify that, we simulate the network with $\tau = [0, 1]$ and show the resulting connectivity in Fig. 6.5(b). Simulation shows that the highest connectivity is attained at $\tau = 0.725$.



(a) (b)

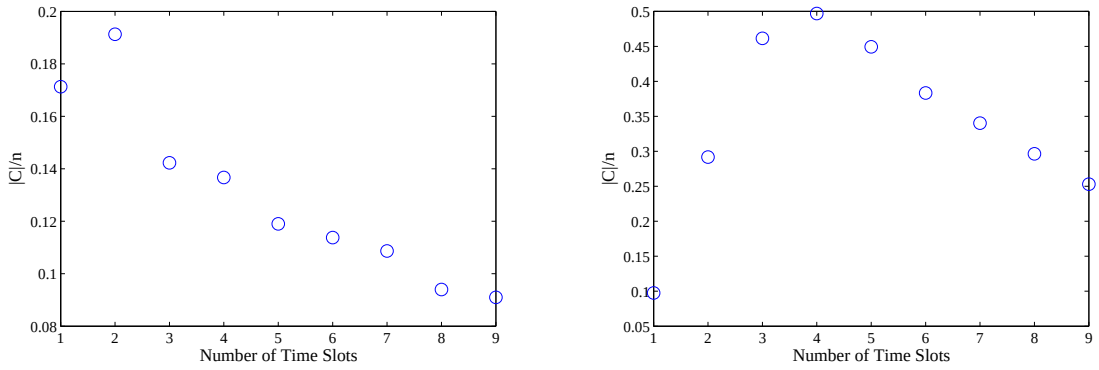
Figure 6.5 (a) $|C|/n$ versus τ for network-1 ($\gamma = 0.08$) (b) $|C|/n$ versus τ for network-2 ($\gamma = 0.06$).

We point out that τ can be used as the ratio of the nodes that are put to i) sleep or ii) powered-off. In both cases $1 - \tau$ will refer to the nodes that are put to transmit or powered-on and the approach will result in same connectivity maximization.

6.1.2.3 Optimal TDMA Slotting

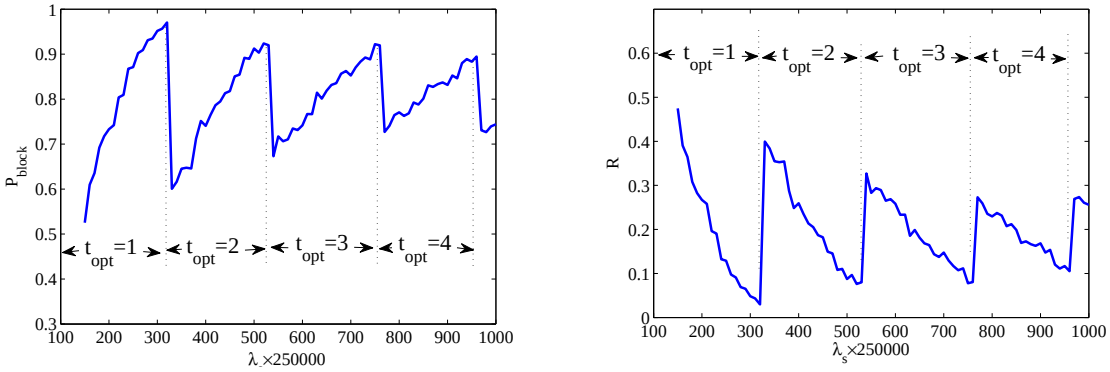
We compare t^{opt} for both networks obtained analytically and via simulations. For network-1 with $\lambda_s = 500/90000$, Eqn. (3.22) results in $t^{opt} = \lfloor 500/300 \rceil = 2$ time slots. To validate this finding we run the simulation from $t = 1$ to 9 time slots and show the resulting connectivity in Fig. 6.6(a). Simulation results show that the maximum connectivity is also attained with 2 time slots.

For network-2 with $\lambda_s = 800/250000$, Eqn (3.22) results in $t^{opt} = \lfloor 800/217 \rceil = 4$ time slots. To validate this finding we run the simulation from $t = 1$ to 9 time slots and show the resulting connectivity in Fig. 6.6(b). Simulation results show that the maximum connectivity is also attained with 4 time slots.



(a) (b)

Figure 6.6 (a) Resultant connectivity for TDMA with time slots from 1 to 9 for network-1 (b) Resultant connectivity for TDMA with time slots from 1 to 9 for network-2.



(a) (b)

Figure 6.7 a) P_{block} vs λ_s under U-MAC. b) Throughput under U-MAC with $t^{opt} = 4$ and $\lambda_s = 800/250000$.

We could not perform any comparative study as there are no other distributed mechanism for characterizing the connectivity maximization of the DSA networks under the SINR model.

6.1.2.4 U-MAC

The goal here is to illustrate the performance of U-MAC. To do so we apply it to network-2 and compare our theoretical findings against the simulation results.

Under $\lambda_p = 0$, we vary λ_s and record the corresponding channel blocking and network throughput as shown in Figs. 6.7(a) and 6.7(b). The zig-zag behavior in P_{block} (and subsequently R), can be understood by examining the number of interferers of a node as shown in Fig. 6.8.

Starting with $\lambda_s = \lambda_s^{opt} = 217/250000$ leads to $t^{opt} = 1$. Note that, with $\lambda_{opt} = 217/250000$, a node will have (on average) 11.5212 interferers. t^{opt} will remain 1 as long as $\lambda_s/\lambda_s^{opt} < 1.5$ i.e., $\lambda_s < 325.5/250000$. As λ_s increases, so does the number of interferers, which increases P_{block} (decreasing R). It can be seen that P_{block} is maximized at $\lambda_s = (1.5 - \epsilon)\lambda_s^{opt} \approx 325.499/250000$ as such density results in the highest number of interferers ($1.4999 \times 11.52 \approx 17$) on that single time slot. An additional increase in λ_s by ϵ leads to $\lambda_s = 1.5\lambda_s^{opt}$; which results into $t^{opt} = 2$ (Eqn. (3.22)). At $\lambda_s = 1.5\lambda_s^{opt} = 325.5/250000$, it can be seen that P_{block} is minimized. This is because such density leads to $(1.5/2) \times 11.52 \approx 9$ interferers for a node, which minimizes the interference and accordingly P_{block}. Such sudden

transition in the number of interferers from maximum to minimum around $\lambda_s = 325/250000$ causes that abrupt max-to-min behavior in P_{block}, and R as shown in Figs. 6.7(a) and 6.7(b).

t^{opt} will remain at 2 time slots as long as $\lambda_s/\lambda_s^{opt} < 2.5 - \epsilon$ (i.e., $\lambda_s < 542.47/250000$). At $\lambda_s = (2.5 - \epsilon)\lambda_s^{opt} \approx 542.4/250000$, the number of interferers will be maximized with $11.52 \times 2.499 \approx 14$ interferer for a node which maximizes P_{block} (minimizing R). Further increase in λ_s by ϵ results in $\lambda_s = 2.5\lambda_s^{opt} \approx 542.5/250000$. Thus, U-MAC will set $t^{opt} = 3$ which minimizes the number of interferers to $11.52 \times 2.5 \times 0.3 \approx 9.599$ for a node which minimizes P_{block} (maximizing R). Again, the sharp transition in the number of interferers from maximum to minimum within an increment in λ_s by ϵ results into that zig-zag behavior again where P_{block} is minimized (maximized) at the same λ_s values which minimizes (maximizes) the number of interferes. The same procedure applies when $t^{opt} = 4, 5, \cdot\cdot, 9$ time slots which gives the zig-zag behavior that is noted in the figures.

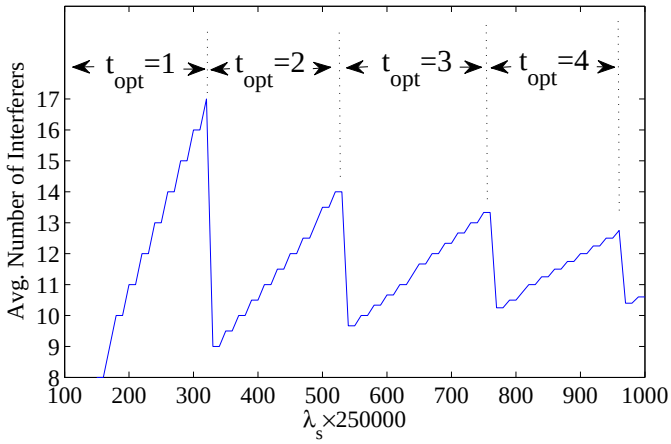
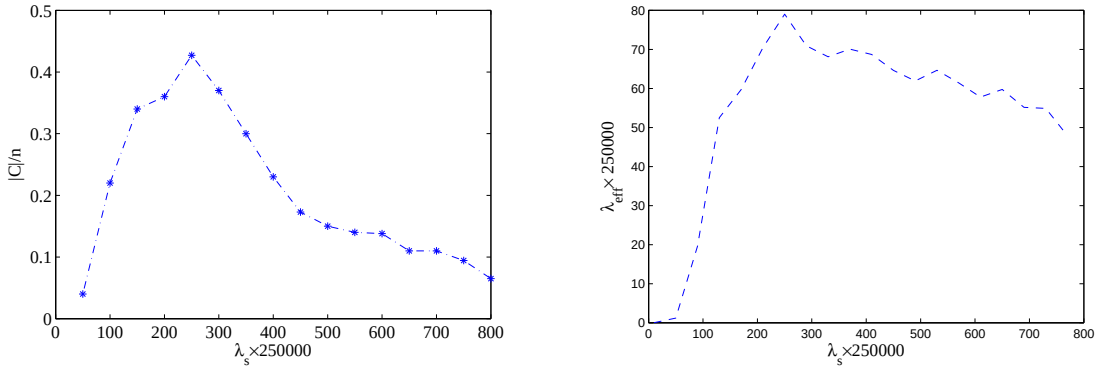


Figure 6.8 Avg. number of neighbors with U-MAC.

Next, we consider the effects of the primary user, were λ_p is set to 4/250000. The resultant connectivity is shown in Fig. 6.9(a). We note that, λ_s^{opt} is achieved at 250/250000 which is relatively higher than $\lambda_s^{opt} = 217/250000$ from the overlay case/setup/scenario. The extra nodes are needed to compensate for the secondary nodes which were evicted by the primary users. Again, with $\lambda_p = 4/250000$, we pick a dense network with $\lambda_s = 800/250000$ and apply U-MAC with $t^{opt} = 3$ as suggested by Eqn. (3.22). Next, the simulation is run with $t = 1$ to 9 time slots and the resulting connectivity is recorded as shown in Fig. 6.10(a). It shows that maximum connectivity is attained with 3 time slots as well. Finally, the throughput is as shown in Fig. 6.10(b).

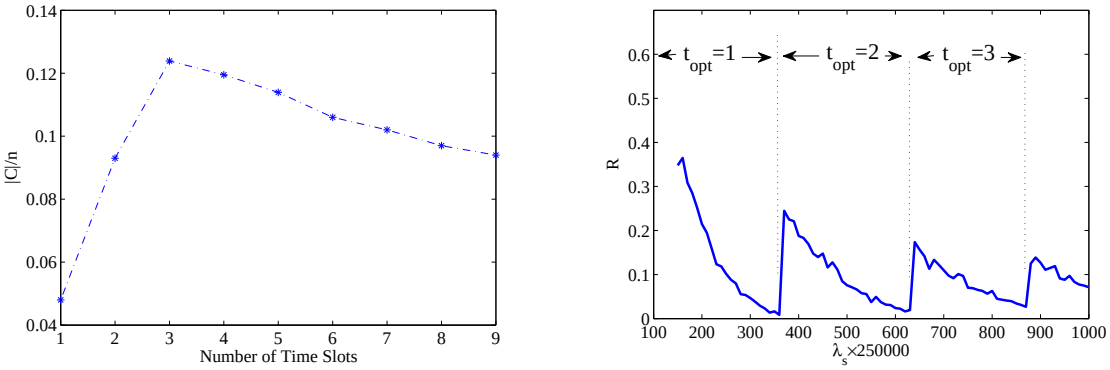


(a) (b)

Figure 6.9 Underlay Mode: a) $|C|/n$ versus λ_s for $\lambda_p = 4/250000$ b) λ_{eff} versus λ_s for $\lambda_p = 4/250000$.

6.1.3 Flooding in DSA Networks Under the SINR Model

We consider a Poisson distributed secondary network with density λ_s over an area of 500 × 500. We use C to denote the size of component that has been flooded. The ratio C/N



(a) (b)

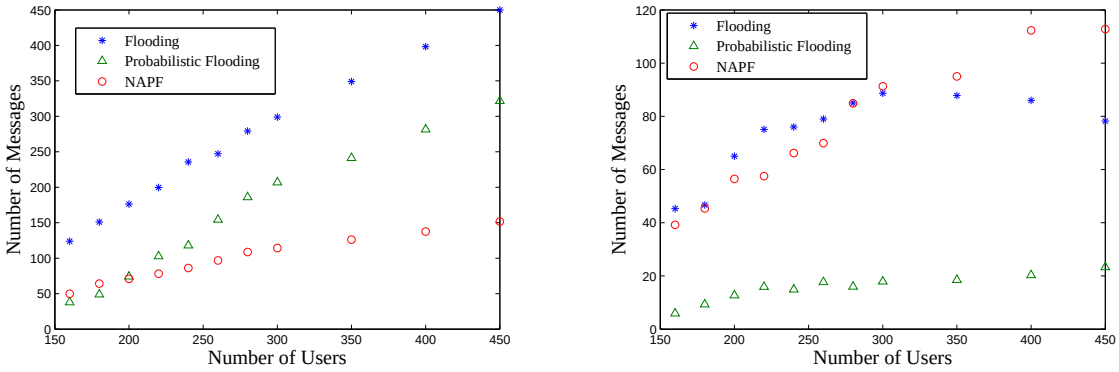
Figure 6.10 Underlay Mode: a) Resultant connectivity of the secondary network with $\lambda_s = 800/250000$ and $\lambda_p = 4/250000$ for TDMA with 1 to 9 time slots. b) P_{block} vs λ_s under U-MAC.

denotes the relative size of the flooded component where N is the total number of deployed secondary users, $C/N = 1$ indicates all the nodes has been flooded/reached. Flooding is initiated by a node (source node) that is placed in the middle of the deployment area.

In order to isolate the combined effects of interference, primary presence, and spatial distribution of the neighbors, we conduct our experiments in two phases. In Phase 1, we consider no primary users, i.e., $\lambda_p = 0$. In Phase 2, we consider primary users, i.e., $\lambda_p > 0$. For both the phases, we compare the performance of NAPF with traditional flooding and probabilistic flooding for the i) Boolean model and ii) SINR model.

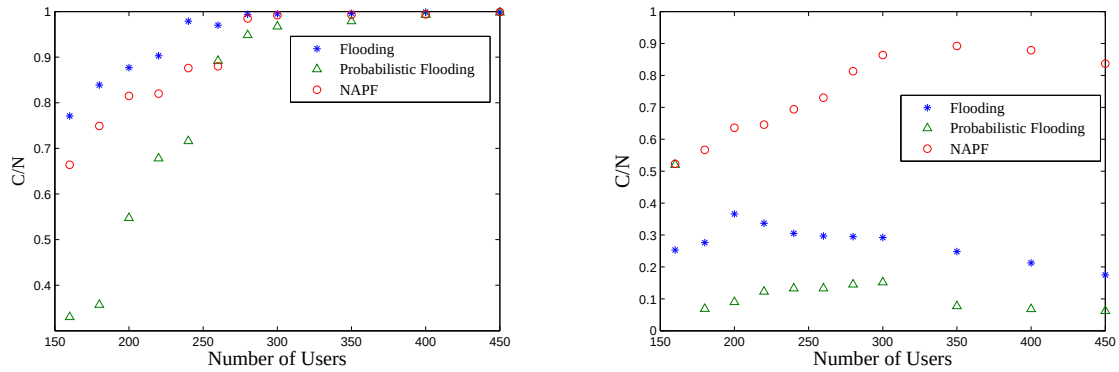
To ensure a fair comparison between NAPF and probabilistic flooding, we use $P = \lambda_c/\lambda_s$ in probabilistic flooding. This is because, NAPF will aim at maintaining $\lambda_c \pi r_s^2$ broadcasting neighbors for each node. Thus, the two techniques will end up (on average) with the same number of rebroadcasting neighbors and the only way NAPF is different from proba-

bilistic flooding is the way it allocates its broadcasting neighbors/group leaders as opposed to the random manner in probabilistic flooding.



(a) (b)

Figure 6.11 (a) Number of Broadcasted messages for the three schemes with no primary and under Boolean model. (b) Number of Broadcasted messages for the three schemes with With no primary and under SINR model.



(a) (b)

Figure 6.12 (a) Resultant connectivity for the three schemes with no primary and under Boolean model. (b) Resultant connectivity for the three schemes with no primary and under SINR model.

6.1.3.1 Phase 1: Primaries Absent

We vary secondary user/nodes per unit area (λ_s) from 160/250000 to 450/250000.

Boolean Model: For this model, we consider $r_s = 50$ for all nodes. Fig. 6.11(a) illustrates the total number of rebroadcasted messages by each technique in response to the broadcast from the source node. The corresponding outreach is shown in Fig. 6.12(a). As expected, conventional flooding achieves the best connectivity/outreach but with the highest number of rebroadcasts. On the other hand, probabilistic flooding does a better job in reducing the number of broadcasted messages while maintaining a relatively good connectivity. Finally NAFP, utilizes a small number of rebroadcasts to achiever similar results by assigning rebroadcast probability to the group leader in each group. In all three cases, the deployment density increases the number of broadcasts needed to cover the network.

SINR Model: For this model, we consider $\gamma = 1$ and $\beta = 1$. Fig. 6.11(b) shows the total number of rebroadcasted messages, while corresponding outreach is shown in Fig. 6.12(b). From Fig. 6.11(b), it can be seen that although conventional flooding uses many rebroadcasts, most of them are unsuccessful as the nodes interfere with each other– the result of which is reflected in the drastically reduced node outreach. Probabilistic flooding generates the worst outreach because the nodes that manage to receive correctly ($SINR \geq \beta$), can decide not to rebroadcast to their neighbors with probability $1 - P$. This results in further thinning of the secondary nodes which are already thinned due to interference– an outcome supporting Lemma 3.2. NAPF on the other hand groups the neighbors and allows only the group

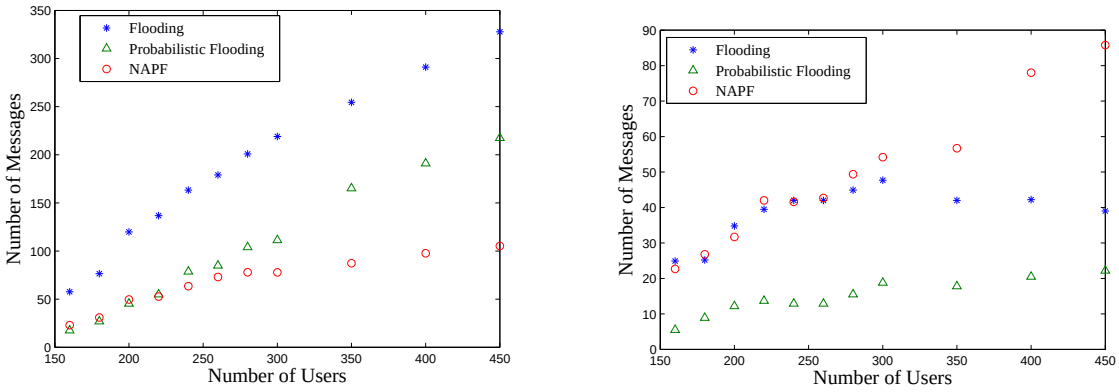
leaders to transmit as per Eqn. (3.29). Note that, such a rule decreases the rebroadcasting probability for the group leader despite an increase in λ_s; thereby reducing the interference at the receivers.

6.1.3.2 Phase 2: Primaries Present

Here, we keep the same density for the secondary nodes, and set the primary density (λ_p) at 7/250000. We also set $r_p = 60$.

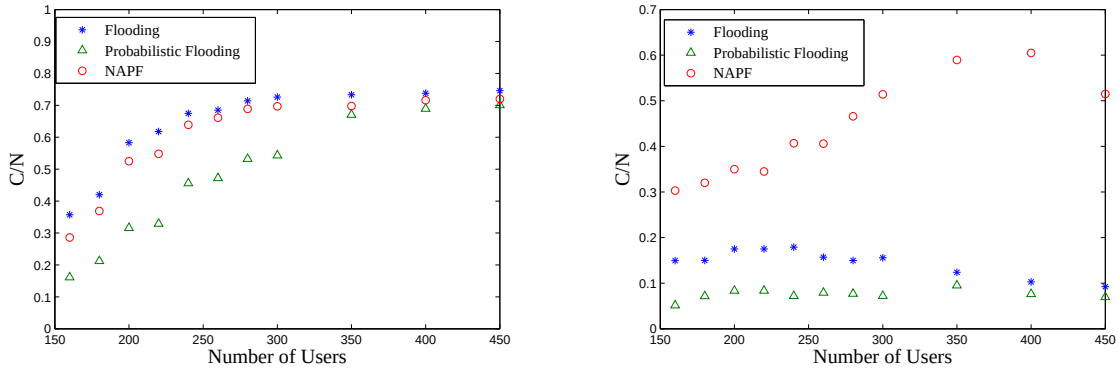
Boolean Model: Fig. 6.13(a) illustrates the total number of rebroadcasted messages by each technique in response to the broadcast of the source node, while corresponding outreach is shown in Fig. 6.14(a). The results show the same behavior as with the primaries absent. However, it can be seen that the values of C/N are smaller than the corresponding ones from Fig. 6.11(a). This is because, all the nodes are not allowed to rebroadcast because of the interference tolerance of the primary users.

SINR Model: Fig. 6.13(b) illustrates the total number of rebroadcasted messages by each technique in response to the broadcast of the source node, while corresponding outreach is shown in Fig. 6.14(b). This scenario exhibits similar behavior to the scenario with no primary users under the SINR model. However the values of λ_s at which C/N maximizes and starts to decline are a little shifted because some nodes (interferes) are prevented from transmission by the primary users. This lowers the total noise level in the system. Thus, it take more nodes to add up the interference to the point where C/N starts to decline.



(a) (b)

Figure 6.13 (a) Number of Broadcasted messages for the three schemes with primary presence and under Boolean model. (b) Number of Broadcasted messages for the three schemes with with primary presence and under SINR model.



(a) (b)

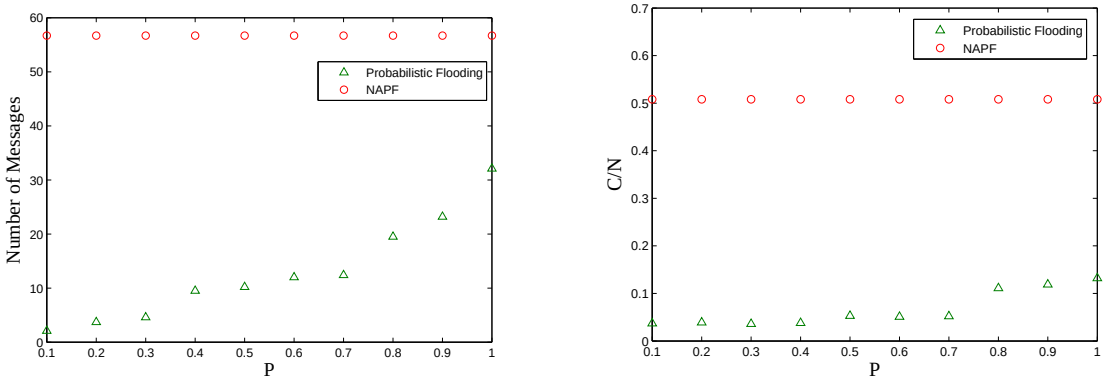
Figure 6.14 (a) Resultant connectivity for the three schemes with primary presence and under Boolean model. (b) Resultant connectivity for the three schemes with primary presence and under SINR model.

6.1.3.3 NAPF vs. Probabilistic Flooding

In the previous sections, the value of P used in probabilistic flooding was calculated with respect to NAPF such that the same number of rebroadcasting neighbors is obtained in both techniques, allowing a fair comparison between both. In this section, we set the value of P

for probabilistic flooding from 0.1 till 1 i.e., independently from NAPF. Also, we consider the SINR model with $\lambda_s = 350/250000$ and $\lambda_p = 7/250000$.

The resultant performance is shown in Figs. 6.15(a) and 6.15(b). From these figures, it can be seen that the performance metrics for NAPF does not depend on P. This is because, P is changed only for probabilistic flooding as NAPF uses the proposed technique to calculate the rebroadcast probability for each node individually. Also, it can be seen that for low values of P, the thinning is severe which results in relatively small values of C/N. As P increases, more nodes participate in rebroadcasting the message; thus both C/N as well as the number of messages increase. At $P = 1$, we see a correspondence with the results of conventional flooding in Figs. 6.13(b) and 6.14(b) for 350 secondary users.



(a) (b)

Figure 6.15 (a) Number of broadcasted messages for NAPF and probabilistic flooding with primary presence and under SINR model. (b) Resultant connectivity for NAPF and probabilistic flooding with with primary presence and under SINR model.

6.1.4 Percolation in Multi-Channel DSA Networks

In this section, we examine the relation between N, γ, λ_p and the percolation of the multi-channel secondary network. We use $|C|$ to denote the size of the biggest connected component and S denotes the total number of deployed secondary nodes. The ratio $|C|/S$ denotes the *relative* size of the biggest connected component. We use the word percolation to refer to the formation of the giant component which contains at least half of the secondary nodes i.e., $S/2 \leq |C_{max}| \leq S$, where C_{max} denotes the size when C has at least half the nodes of S. Thus when percolation occurs, we get $0.5 < (\theta_p = |C_{max}|/S) \leq 1$. We present 2 scenarios and discuss the results accordingly.

Table 6.1 Theoretical and practical values of N^{opt}, N^L and N^U for $\gamma = 0.01,\ 0.1$, and 0.5. ϕ implies does not exist.

γ	Theoretical Values			Practical Values		
	N^{opt}	N^L	N^U	N^{opt}	N^L	N^U
0.01	9	ϕ	41	9	ϕ	44
0.1	13	7	42	12	6	41
0.5	30	ϕ	ϕ	28	ϕ	ϕ

6.1.4.1 Scenario 1: $\lambda_p = 0$

In this scenario, we illustrate the results of Proposition 3.1. We deploy 340 secondary nodes in a square region of 400×400 following a Poisson distribution with density $\lambda_s = 340/(400\times400)$ nodes per unit area. M is set to 5. r_s and r_I are 25 and 35 respectively. P is set to 1 Watt and $\alpha = 2$. Via simulations, λ_s^c was found to be 0.0018125 nodes per unit area. Three values

of γ are considered (0.01, 0.1 and 0.5) which correspond to $K = 24, 9$, and 3 respectively (obtained via simulation). As defined earlier, K is the average number of interferers a receiver can tolerate before it is interfered. The resulting connectivity for each case is shown in Figures 6.16, 6.17 and 6.18. From these 3 figures, the practical values of N^{opt}, N^L and N^U are obtained and compared to the theoretical values obtained by solving Eqns. (3.38) and (3.39). The comparison is shown in Table 6.1.

For $\gamma = 0.01$ with $\lambda_s = 0.002125$ the expected number of interferers of each transceiver is approximately 9. This is less than the number of interferers it can handle, so effects of interference are negligible. For that reason, there exists no value for N^L, both practically and analytically, because the network is already percolated at $N = 1$. As N exceeds $2M$, channel abundance occurs which thins the secondary network. Thinning increases as N increases and

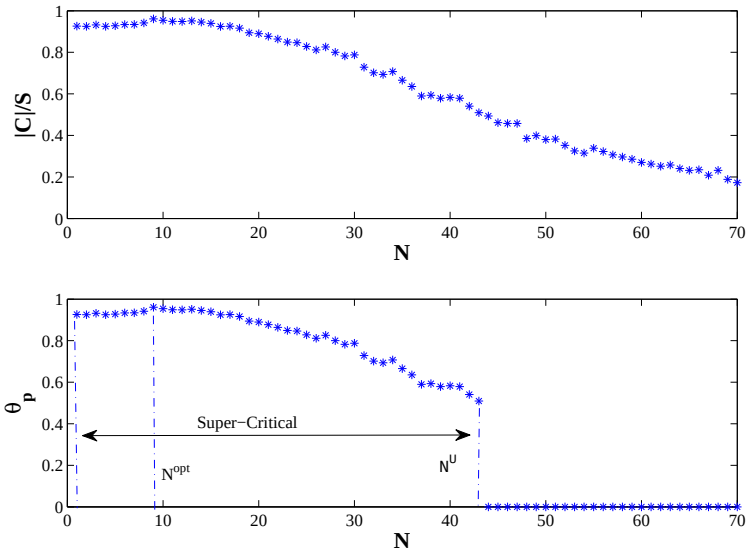


Figure 6.16 $|C|/S$, θ_p versus N for $\gamma = 0.01$.

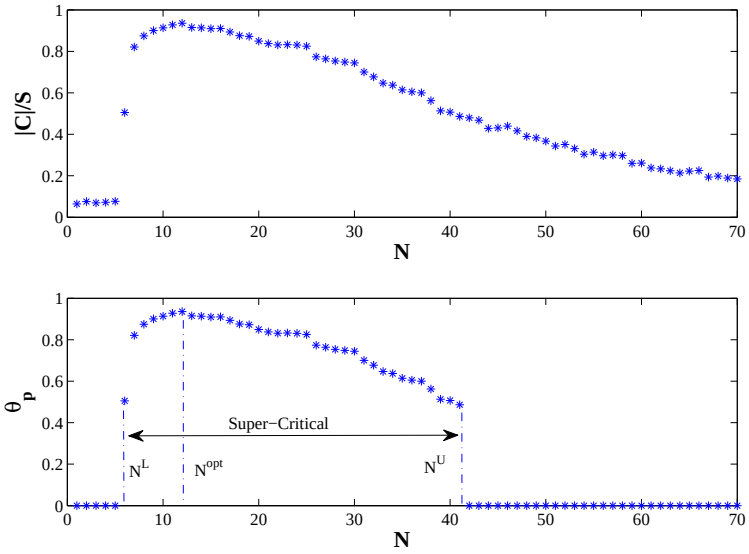


Figure 6.17 $|C|/S$, θ_p versus N for $\gamma = 0.1$.

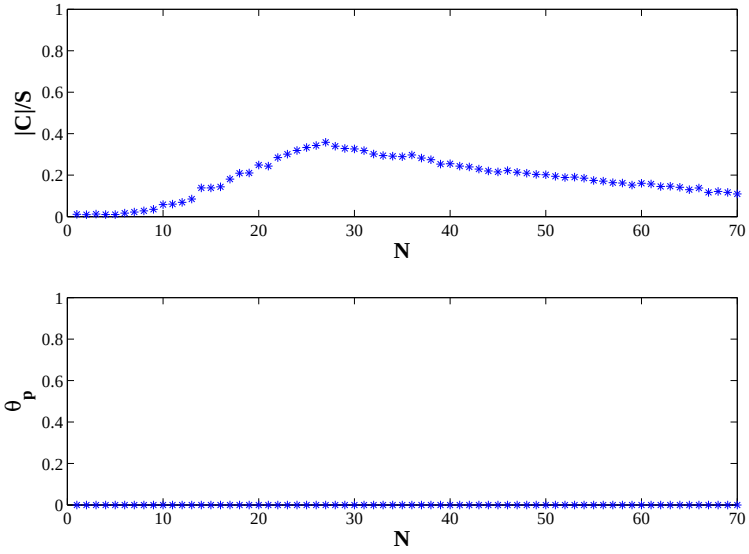
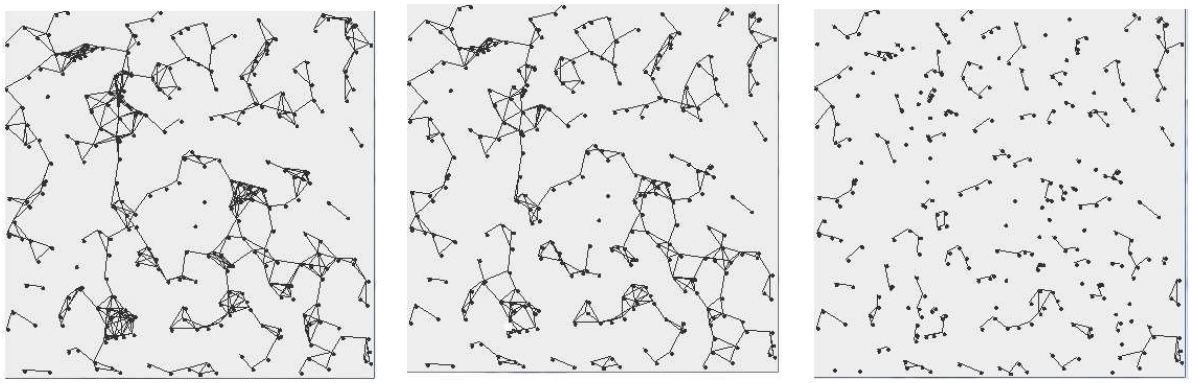


Figure 6.18 $|C|/S$, θ_p versus N for $\gamma = 0.5$.

at $N = 44$, the network becomes too thinned to be percolated as seen in Fig. 6.16 As a result, $|C|/S$ drops below 0.5.



(a) (b) (c)

Figure 6.19 340 nodes, $N = 6$, and $M = 5$ (a) $\gamma = 0.01$, $|C|/S = 81.3\%$ (b) $\gamma = 0.1$, $|C|/S = 61.4\%$ (c) $\gamma = 0.5$, $|C|/S = 3.2\%$. • represents a secondary user/transceiver, and — represents a two-way communication link.

For $\gamma = 0.1$ and $N < 6$ the network is already thinned due to interference (interference dominated region). As N increases to 6, the neighboring nodes are guaranteed to have a common channel. However, the interferers become scattered over more channels. This reduces P_{tx} from 1 to 5/6 resulting in percolation for the first time. As N increases so does θ_p and $|C|/S$. The optimal (maximum) connectivity is attained at $N = 12$ as seen in Fig. 6.17. Further increase in N makes thinning due to interference negligible but it makes channel abundance thinning dominant. Thus θ_p starts to decrease until the network is no longer connected for $N > 40$.

For $\gamma = 0.5$, the network is heavily thinned due to interference. Increase in N reduces that interference-effect until the maximum connectivity is attained at $N = 28$. However, even at $N = 28$ (N^{opt}) the network did not percolate ($|C|/S < 0.5$). Thus under the current network parameters, no value of N can minimize the interference and channel-abundance thinning such that the network would percolate.

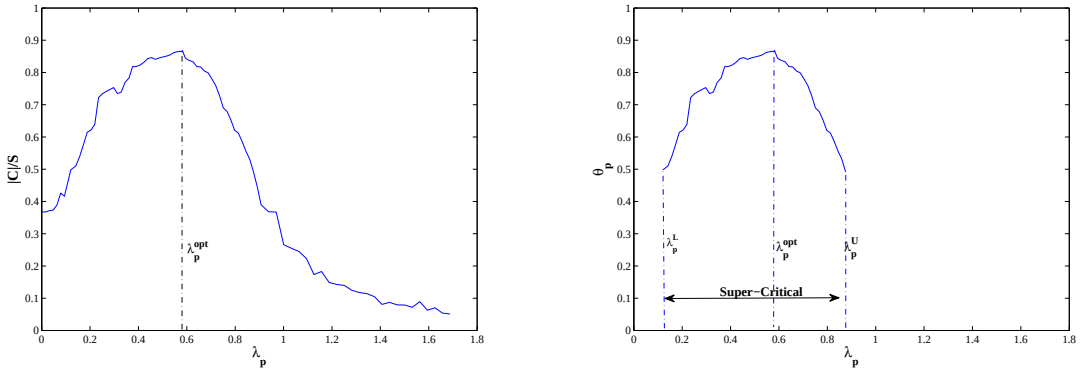
Figs 6.16, 6.17 and 6.18 bound the range of N for which the network percolated. Interestingly, Fig. 6.18 shows no such ranges as the network never percolates. The effect of varying γ is further illustrated in Figs. 6.19(a), 6.19(b) and 6.19(c). We consider a network with 340 nodes and set $N = 6$ and $M = 5$. We use the same values γ as before. With $\gamma = 0.01$, the network has the most number of links. With increased interference (i.e., $\gamma = 0.1$), the number of links decreases. There are hardly any links for $\gamma = 0.5$.

6.1.4.2 Scenario 2: $\lambda_p > 0$

We illustrate Lemma 3.5 by showing the effects of λ_p on the connectivity of the secondary network. We use the same secondary network that was used in scenario 1 with $\gamma = 0.1$ (also with the same setup parameters). Unlike scenario-1 where N was a variable, we fix it at 50. From Fig. 6.17 we note that, for $N = 50$ the secondary network does not percolate with $|C|/S = 0.367$ and the secondary network is in the channel-abundance-thinning dominated region. For increasing values of λ_p, the corresponding connectivity and θ_p are shown in Figs. 6.20(a) and 6.20(b) respectively.

In Fig 6.20(a), we note that, increase in λ_p from 0 to λ_p^{opt} is accompanied by an increase in $|C|/S$. This is because the arrival of the primary users reduces the set of available channels for the secondary users. This in turn reduces P_{thin} which increases the connectivity because the secondary network is in the channel-abundance-region. From Figs. 6.20(a) and 6.20(b), it can be seen that percolation first occurs at $\lambda_p^L = 0.12$ (i.e., total of 192 deployed nodes).

θ_p keeps increasing with λ_s until the maximum connectivity is attained at $\lambda_p^{opt} = 0.583125$ (i.e., total of 933 deployed nodes). Further increase in λ_p pushes the nodes into the channel-deprivation region i.e., the number of available channels has shrunk such that more and more transceivers are ending up on the same free channels; thus increasing the interference experienced by each receiver. At this point, the connectivity starts to decrease due to interference until the network is no longer percolated which occurs at $\lambda_p^U = 0.87875$ (i.e., total of 1406 deployed nodes). Another factor which reduces the connectivity is that some secondary nodes will end up with no channels because of the primary users. The theoretical values for λ_p^L, λ_p^{opt} and λ_p^U are 0.08625 (i.e., 138 nodes), 0.618125 (i.e., 989 nodes) and 0.918125 (i.e., 1469 nodes) respectively.



(a) (b)

Figure 6.20 A) $|C|/S$ versus λ_p for $\gamma = 0.1$; B) θ_p versus λ_p for $\gamma = 0.1$.

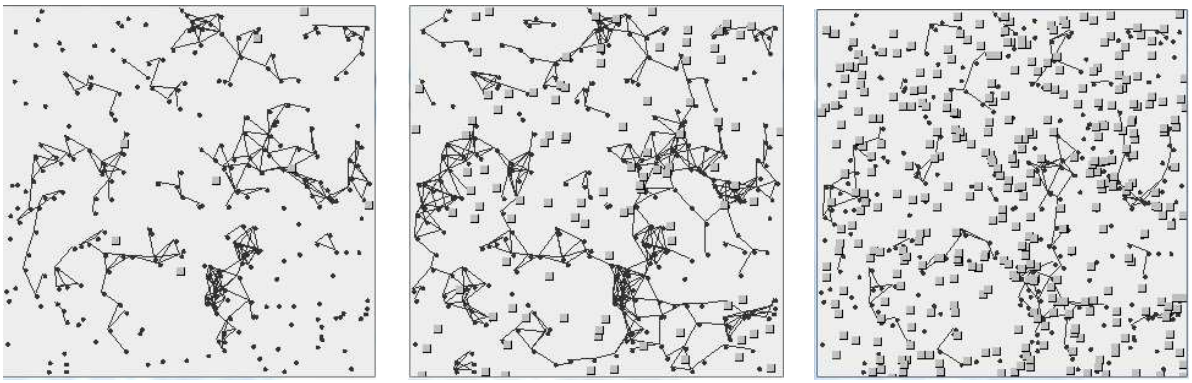
6.1.4.3 Cooperation between Primaries and Secondaries

Following the results of Lemma 3, we argue that primary-secondary cooperation can increase the connectivity of both. As for the primary network, cooperation can be achieved by controlling r_p or λ_p, or both. The idea is to add enough number of primary users to eliminate the channel abundance or remove primary users which increases the number of vacant channels for the secondary users. For both cases (adding, removing primary users), we choose λ_p such that $\lambda_p = \lambda_p^{opt}$, which drives the connectivity of the secondary network to the maximum. As for the secondary network, cooperation is achieved by relaying data on behalf of primary users as was shown in [25].

Through Figs. 6.21(a), 6.21(b) and 6.21(c), we demonstrate how we can achieve cooperation via controlling λ_p. We consider the network setup used in Scenario 2 with a total of 340 static secondary deployed transceivers. For λ_p, we use $\lambda_p = 0.00625,\ 0.538$ and 0.947. In Fig. 6.21(a), $\lambda_p = 0.00625$, the network is in channel abundance with $|C|/S = 33.1\%$. Note that, not many links are established in-spite of nodes being in range. On the other hand, in Fig. 6.21(c) the network is in channel deprivation region with $\lambda_p = 0.947$ and $|C|/S = 5.31\%$. Now if the primary users agree to adjust λ_p such that $\lambda_p = \lambda_p^{opt}$, it results in the network shown in Fig. 6.21(b). Such λ_p value drives θ_p into the optimal point where $\theta_p = 87.51\%$.

Obviously in Fig. 6.21(b), there are less number of primary users than in Fig. 6.21(c) which should decrease their coverage area. However, at the optimal point (Fig. 6.21(b)),

the secondary coverage is maximized which helps improve the primary's coverage area by relaying the primary's data.



(a) (b) (c)

Figure 6.21 Illustration for the secondary users connected component(s) for: (a) $\lambda_p = 0.00625$ (channel abundance), (b) $\lambda_p = 0.538$ (optimal point) (c) $\lambda_p = 0.947$ (channel deprivation). $\square$ represents a primary user, $\bullet$ represents a secondary user/transceiver, and —— represents a two-way communication link.

6.2 Capacity

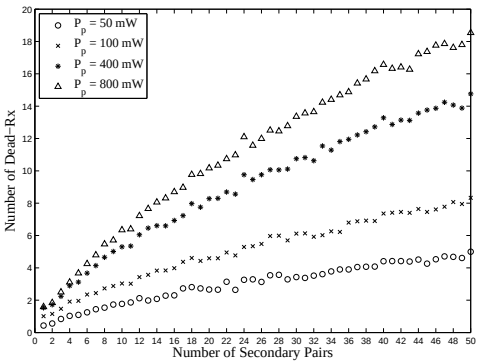
6.2.1 Capacity Bounds and Optimizations

To verify the validity of the theoretical findings, we conducted simulation experiments where we considered a randomly deployed secondary DSA network in a square region of 100×100 units with two primary transceivers (forming a transmitter-receiver pair) at the center. We considered various network sizes by varying the number of secondary transmitter-receiver pairs from 1 to 50. The transmitters and receivers were paired using exhaustive search pairing. Total bandwidth (B) was 1 MHz. Power of the primary transmitter is set to 800

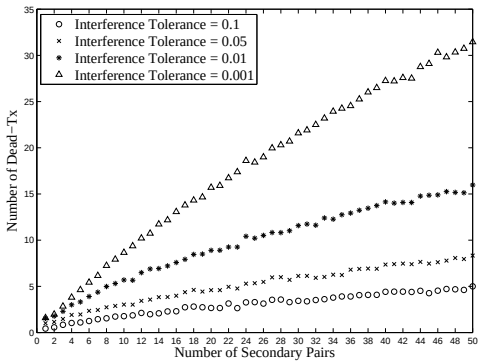
mW, while the interference tolerance is set to 0.1 W at the primary receiver. The secondary receiver threshold (β) was set to 1. Path loss exponent (α) was 2. No ambient noise (N_0) was considered in obtaining the dead-Tx and dead-Rx nodes, to ensure that any dead node was due to the effects of the primary only. Gaussian noise with N_0=0.001 W was considered for capacity simulations.

6.2.1.1 Dead-Tx, Dead-Rx

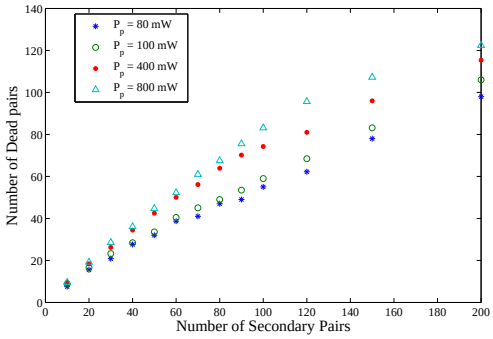
To study the effects of the primary transmitter, 4 different power levels [50 mW, 100 mW, 400 mW, 800 mW] for the primary transmitter were used. The number of secondary pairs (n) was incremented 1 pair at a time, from 1 to 50. For each value of n, we generated 200 different random topologies so as to average out any topological effects. For every topology, we check for dead-Rxs using Eqn. (4.16). In Figure 6.22(a) we see how the number of dead-Rxs increases with increasing network size. This is because as more nodes are added randomly in a fixed area, more nodes get deployed in the vicinity of the primary transmitter, thus increasing the dead-Rx count. More dead nodes means less number of pairs' requests are considered in the input power vector which reduces the search space of the optimizer. For example, in Figure 6.22(a) with primary power of 800 mW, it can be seen that on the average 18 out of the 50 pairs are dead-Rx. That is, using Eqn. 4.16 we managed to reduce the dimensional state space from 50 to 32 dimensions.



(a)



(b)



(c)

Figure 6.22 (a) Number of dead receivers (b) Number of dead transmitters (c) Number of dead pairs with 10 primary transceivers, $\gamma = 0.1$, and 4 different power levels for the primary transceivers.

To study the effects of the interference tolerance/threshold condition, we used 4 different values of γ [0.1 W, 0.05 W, 0.01 W, 0.001 W]. The primary transmit power was set at 200 mW. The number of dead transmitters is shown in Figure 6.22(b). Just like the dead-Rxs, as the deployment density increases the number of dead-Txs also increases. This is because the number of secondary transmitters in the vicinity of the primary transmitter (i.e., in the dead-Tx circle) increases. Just like the dead-Rx reduced the search space, dead-Tx does exactly the same thing. For example, in Figure 6.22(b) with $\gamma = 0.001$ using

Eqn 4.24, on average we reduced the optimizer's search space from 50 to 18 dimensions and with $\gamma = 0.01$, the reduction was from 50 to 34 dimensions.

6.2.1.2 Combined Elimination Schemes

In this section we filter the input power vector using all the elimination schemes which we pointed out earlier: dead-Rx (Eqn. 4.16), dead-Tx (Eqn. 4.24), and coverage zones (Eqn. 4.30). The number of secondary pairs ($n/2$) was changed from 0 to 200 pairs (400 nodes) with 10 and 20primary transceivers. Primary and secondary nodes were randomly deployed.

Figure 6.22(c) shows the dead pairs when 10 primary transceivers are deployed with $\gamma = 0.1$. Same level of transmission power P_p is used in all of the primary transceivers. 4 different power levels of [50 mW, 100 mW, 400 mW, 800 mW] are used for P_p. For this case, in order to illustrate the effects of dead pairs, we have increased the number of secondary pairs up to 200 pairs. Interestingly for $P_p = 800$ mW, using all the elimination schemes, on average the dimensions of the search space are reduced from 200 to $200 - 122 = 78$ while for $P_p = 80$ mW, the reduction is from 200 to 102.

Next, the number of primary transceivers is increased to 20 transceiver and the number of dead-pairs is recorded as show in Figure 6.23(a). Notably for $P_p = 800$ mW, using all the elimination schemes the dimension of the search space is reduced from 200 to $200 - 182 = 18$ while for $P_p = 80$ mW, the reduction is from 200 to 63.

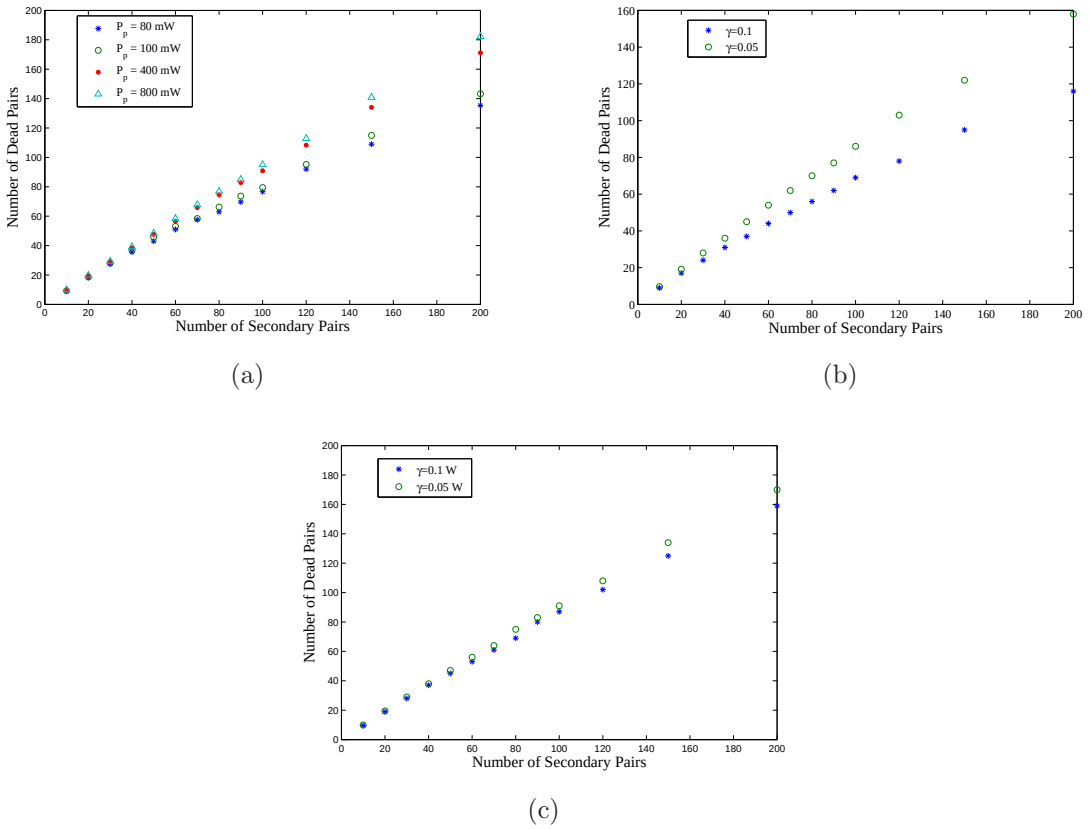


Figure 6.23 (a) Number of dead pairs with 20 primary transceivers, $\gamma = 0.1$, and 4 different power levels for the primary transceivers (b) Number of dead pairs with 10 primary transceivers, $P_p = 200$ mW, and 2 different values for γ (c) Number of dead pairs with 20 primary transceivers, $P_p = 200$ mW, and 2 different values for γ.

Now, we set the number of primary transceivers to 10 again and use 2 different values for γ. The primary transmit power P_p was set at 200 mW. All the primary transceivers use the same values for γ and P_p .

Figure 6.23(b) shows the dead pairs when 10 primary transceivers are deployed with $P_p = 200$ mW, γ is set to 0.1 in one scenario and 0.05 in the other. From Figure 6.23(b) we can see the effects of γ on the number of dead-pairs and consequently on dimensionality reduction.

In Figure 6.23(c), the number of the primary transceivers is set to 20 and the number of dead-pairs is also recorded for γ of 0.1 and 0.05. Obviously more pairs are dead since the additional primary transceivers impose further constraints on the maximum power of each pair resulting in more dead pairs.

Notably, in all the four Figures 6.22(c), 6.23(a), 6.23(b), and 6.23(c) for small values of $n/2$ (which correspond to sparse deployments) we see that the number of dead pairs is approximately equal to the number of deployed pairs. As the density (number of pairs) increases, more pairs become alive. The reason for that is: as the density increases, the transmitter-receiver pairs are becoming closer to each other which is translated to higher SINR values.

6.2.1.3 Relative SINR Goodness ($SINR^{Rel}$)

In order to demonstrate the proposed metric for capacity maximization goodness, we simulated a network with 34 transmitter-receiver pairs distributed around a primary transceiver with P_{p_1}= 800 mW and γ=0.1. The $SINR^{Rel}_{\mathcal{T}_j,\mathcal{R}_i}$ for all the pairs are found as shown in Fig. 6.24(a). When the pairs are arranged in descending order of their $SINR^{Rel}_{\mathcal{T}_j,\mathcal{R}_i}$ values, they result in the ordered set $\mathcal{U} = \{(\mathcal{T}_3, \mathcal{R}_3)$, $(\mathcal{T}_6, \mathcal{R}_6)$, $(\mathcal{T}_{18}, \mathcal{R}_{18})$, $(\mathcal{T}_{17}, \mathcal{R}_{17})$, $(\mathcal{T}_{28}, \mathcal{R}_{28})$, $\cdots$, $(\mathcal{T}_2, \mathcal{R}_2)\}$ as shown in Figure 6.24(a).

If the maximum capacity is to be found in $\mathcal{O}(1)$, the entire γ has to be allocated to a single pair; that pair is U_1 i.e., pair number 3 ($\mathcal{T}_3$-$\mathcal{R}_3$) as shown in Figure 6.24(a). The

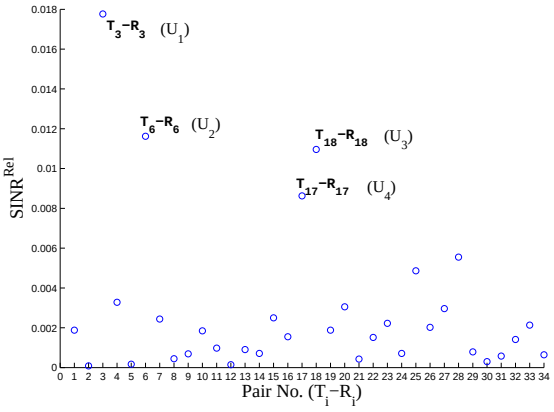
maximum capacity using 2 pairs only can be attained by allocating γ to U_1 and U_2 which corresponds to pair number 3 and pair number 6 ($\mathcal{T}_3$-$\mathcal{R}_3$, $\mathcal{T}_6$-$\mathcal{R}_6$). Continuing, the maximum capacity for K pairs was found by allocating the power budget to the first K entries in $\mathcal{U}$. The capacities obtained for $K = [1, |\mathcal{U}|]$ pairs taken from the first K elements in $\mathcal{U}$ are shown in Fig. 6.24(b) with $|\mathcal{U}| = 34$ pairs.

6.2.1.4 Optimal and Sub-optimal Solutions

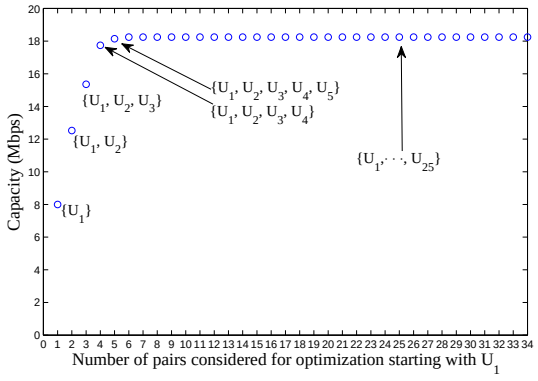
First, we show how to perform the optimization with the reduced search space without compromising optimality. Then, we discuss the trade-off for achieving a sub-optimal solution versus optimization over smaller search spaces.

Optimal Solution: For the 34 pairs whose $SINR^{Rel}$ values are shown in Fig. 6.24(a), the number of dead pairs is 9– thereby immediately reducing the search space from 34 variables (dimensions) to 25. However, we still managed to retain the global maximum via optimizing the capacity over a smaller number of variables without compromising the optimality of the solution. For that we propose two approaches: linear and binary searches.

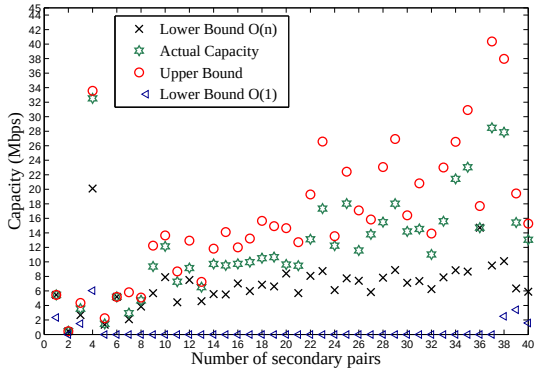
- Linear Search: The idea is to start to compute the system capacity starting with the first pair (U_1) in the ordered set $\mathcal{U}$ and adding one pair at a time (the pairs are taken in order from $\mathcal{U}$). When adding a pair does not increase the capacity any more, the process is stopped.



(a)



(b)



(c)

Figure 6.24 (a) $SINR^{Rel}$ for all 34 pairs; (b) Resultant maximum capacity K=[1,34] pairs. Pairs are selected with descending $SINR^{Rel}$; (c) C_{UB}, C_{LB}, and the actual maximum capacity for 40 randomly deployed networks.

We started with $\mathcal{U}_1$ and found the corresponding capacity $C_1 = 8$ Mbps as shown in Fig. 6.24(b). Then $\mathcal{U}_2$ was added and the capacity C_2 of U_1 and U_2 was found to be 12.52 Mbps. Since $C_2 > C_1$ it means convergence did not occur. Next U_3 was added and C_3 was also found and compared against C_2. The procedure was repeated until convergence was obtained at U_6 i.e., the maximum capacity was obtained by optimizing over $\{U_1, \cdots, U_6\}$ instead of optimizing over $\{U_1, \cdots, U_{25}\}$. To check the integrity of our solution, we kept on including more pairs from $\mathcal{U}$ and finding the maximum capacity as shown in Fig. 6.24(b). The figure clearly shows that incorporating $\{U_7, \cdots, U_{34}\}$ with $\{U_1, \cdots, U_6\}$ did not change the maximum capacity that was obtained with $\{U_1, \cdots, U_6\}$. It is obvious that, optimizing over only 6 pairs is much faster than optimizing over the 34 pairs given that the two optimizations produced the same result.

Note that the number of nodes to be incorporated in each step can be set according to the user preferences. The same applies to the number of the nodes to start with. Technically, the number of start-up nodes and the step-size control the convergence time as well as the optimality of the optimization.

- Binary Search: Using binary search, we consider the first $\lceil (25/2) \rceil = 13$ entries of $\mathcal{U}$ i.e., (U_1 to U_{13}). The maximum capacity that resulted from 13 pairs was 18.152 Mbps. Next, we considered the first $\lceil (13/2) \rceil = 7$ entries of $\mathcal{U}$. That resulted in 18.152 Mbps as well. This meant that pairs U_8 to U_{34} did not contribute to the maximum capacity. We continue the binary split and use the first 4 entries of $\mathcal{U}$; the resultant capacity was 18.0931 Mbps. Since $18.0931 < 18.152$, it means that some of the pairs between

U_5 and U_7 contributed to the capacity. Further investigations reveal that the first 6 entities in $\mathcal{U}$ resulted in the maximum capacity of 18.152 Mbps. This can be seen in Fig. 6.24(b) where $U_1, \cdots, U_6$ attain the maximum capacity of 18.152 Mbps beyond which there is no improvement, i.e., convergence occurred at U_6.

Sub-optimal Solution: If instead of continuing the linear or binary searches till convergence is reached, we stop after some iterations, then obviously the optimal solution is not obtained. Alternatively, we can choose to avoid the search and simply consider the first K elements of $\mathcal{U}$ i.e., we choose to optimize a K dimensional space, $K < n/2 - n_{dead}$. By doing so, a sub-optimal solution is obtained. For example, in optimizing for $K = 3$ (3-dimensional space) we simply optimize over $\{U_1, U_2, U_3\}$ to get a corresponding capacity of 15.36 Mbps as shown in Figure 6.24(b). That is the highest capacity obtained by optimizing over 3 pairs only.

6.2.1.5 Upper and Lower Capacity Bounds

To study the proposed upper and lower bounds, we simulated 40 random networks of sizes 1 to 40 secondary pairs. Power of the primary transmitter is set to 800 mW, while the interference tolerance is set to 0.1 W at the primary receiver. Gaussian noise with variance of 0.001 is used. The maximum achievable capacity (which we refer to as 'Actual Capacity') for each network of a given size is found using a modified Nelder-Mead Method [112] with

penalization. C_{UB}, $C_{LB,\mathcal{O}(1)}$ and $C_{LB,\mathcal{O}(n)}$ are computed using Eqns. (4.12), (4.14) and (4.15) respectively. The capacity results are shown in Fig. 6.24(c).

Effect of Topology: From Fig. 6.24(c), it can be noted that the capacity does not necessarily increase with network size. For example, the network with 32 secondary pairs resulted in a capacity that is lower than the capacity achieved by the network with 22 pairs. This is because of the pathological topology that was created by 32 nodes. However, on an average, larger networks would yield higher capacities when topological effects are averaged out. Smaller inter-nodal distances means higher $SINR^{max}_{\mathcal{T}_j,\mathcal{R}_i}$ values, which means more pairs would satisfy the power split condition, hence increasing the upper bound and actual capacity of the network.

The network obtained with 6 pairs is shown in Fig. 6.25. Strangely, its capacity and both bounds are equal as seen in Fig. 6.24(c). This is because no pair other than $\mathcal{T}_1$-$\mathcal{R}_1$ qualified for the split condition (Eqn. 4.11). So the entire power was allocated to only one pair ($\mathcal{T}_1$-$\mathcal{R}_1$) which has $SINR^{max}$, resulting in C_{UB}, $C_{LB,\mathcal{O}(n)}$ and the actual capacity being equal. From Fig. 6.24(c) it can be noted that for high values of n (dense deployments) $C_{LB,\mathcal{O}(1)} = 0$. This is because, with more nodes, the probability of a node being too close to the primary increases. Such closeness renders the node dead. Thus with random selection in $\mathcal{O}(1)$, it might so happen that the power is allocated to a dead-pair (dead-tx, dead-rx) resulting in zero capacity. For sparse deployments the chances of being in the vicinity of a primary user is lower which results in $C_{LB,\mathcal{O}(1)} > 0$ as can be seen in Fig 6.24(c). We point out that:

1. The derived bounds are computed in $\mathcal{O}(n)$, which affects their tightness. They can be made tighter by trying combinations of H_{k1} and H_{k2} instead of using H_{k1} only when doing the replacement for the qualified for splitting pairs. However that will take at least $\Omega(n)$. The bounds can be made even tighter by trying combinations of H_{k1}, H_{k2}, and H_{k3} instead of H_{k1} and H_{k2} only, again this will consume even more time. Apparently, The bounds can be made tighter and tighter by trying combinations of more pairs and this is why the problem is NP-hard.

   H_{ki} was defined as an element of the set $\mathcal{H}_k$ in section 4.1.2.1.

2. We did not compare our work with any existing work because all of them either find the asymptotic bounds for the networks i.e., $n \to \infty$ or use power control algorithms that maximize the user's desired throughput rather than exploring the capacity bounds of the network itself.

6.2.2 QoS Evaluation

We would like to demonstrate the feasibility of providing a QoS evaluation platform that can work with real systems. We emphasize that the VQ-based method improves the speed in execution as compared to executing the original capacity equation. Such improvement empowers and opens the horizon for using relatively slower computational devices in computation intensive applications. In this regard, we implement the proposed concept of VQ-based QoS evaluation on an 8-bit micro-controller where we compare the time of executing the original

capacity equation and the look up time using a precomputed codebook. The look up is done through both linear search and binary search.

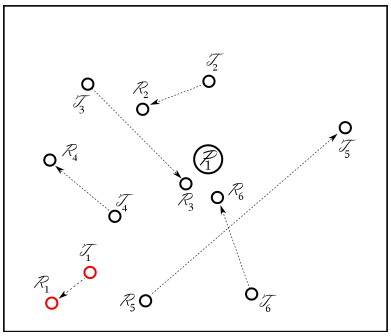


Figure 6.25 Network with 6 pairs where $C_{LB,\mathcal{O}(n)}$ =C_{UB} = actual capacity.

It is to be noted that, the computational complexity of the equation to be evaluated depends on the number and nature of the mathematical operations needed to evaluate the equation. For example, finding the total network capacity of K pairs involves evaluating $\sum_{i=1}^{K} log_2(1 + SINR_i)$. From an embedded hardware point of view, there are two potential problems with the above computation.

1. Finding $SINR_i$ means to evaluate:

$$SINR_i = \frac{P_i/x_i^{\alpha}}{\sum_{j=1, j\neq i}^{K} P_j/x_j^{\alpha}} \tag{6.1}$$

From a micro-controller's point of view this is a time consuming process because it involves multiplication, division, raising to power and finding the logarithm given that: i) number of Million-instructions per second (MIPS) of a micro-controller is in the order of tens MIPS at best, and ii) the cheapest micro-controllers are not equipped

with hardware multiplier modules, thus all the operations have to be computed using additions and subtractions only. We will show that even with micro-controllers that are equipped with hardware multipliers, the process is still time consuming mainly due to the limit set by the MIPS.

2. The computation of the capacity equation involves handling fractional numbers multiple times (for each $\log(\cdot)$ and $SINR_i$ term). It is to be noted that repeated handling of fractional numbers on 8 or 16 bit devices can result in rounding errors even if multi-byte allocation is used.

Such limitations make micro-controllers an infeasible option when considering such computation intensive operations especially if the result has to be calculated within a delay budget. This is why such calculations are usually performed using 24 or 32-bit DSP or micro processors despite the micro-controller being robust, flexible, and most importantly cheaper in cost as compared to microprocessors. Notably 24 (32) bit micro-controllers are still slower than the 24 (32) DSP and micro-processors since they employ `Multiply Accumulate` feature which is not performed by the micro-controllers.

6.2.2.1 Implementation Platform

We implement VQ on two 8-bit devices: PIC18F87J60 and PIC16F877A, the former is running on a crystal oscillator of 25 MHz while the latter is running on a crystal oscillator

of 8 MHz. Fig 6.26 shows the micro-controller development board that was used to in this work along with the PIC18F87J60 and the software development tools.

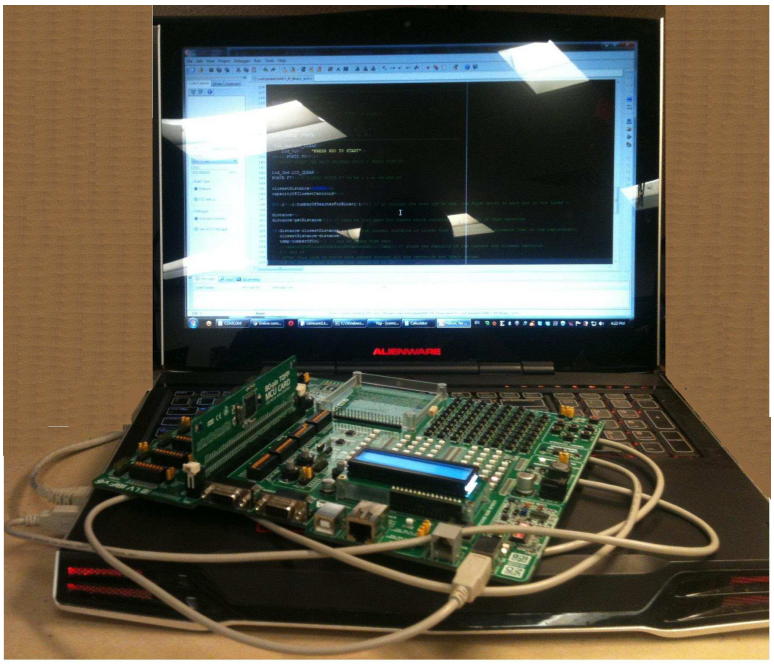


Figure 6.26 Implementation testbed.

PIC18F87J60 is equipped with a hardware multiplier module while PIC16F877A has no hardware multiplier module. Thus any mathematical operation that involves multiplication (like log) will consume more execution time on the PIC16F877A which further lowers the maximum value of K in Eqn. (6.1) for a given time budget.

From the above discussion, it is clear that scalability (with respect to K) is an issue with this embedded hardware platform if it was to evaluate the capacity equation as it is. On the other hand VQ is scalable because the centroids/code-words are precomputed and then stored in the micro-controller. Finally, they will be looked up against the input power

vector to find the best match. By doing so, the execution speed will be greatly improved since the values will be looked up instead of being computed.

6.2.2.2 Implementation Results

We primarily focus on how the VQ algorithm performs with respect to time as compared to evaluating the capacity equation as both of them are run on the PIC16F877A and PIC18F87J60. The results were obtained as follows:

1. For a given K, the capacity equation was computed for the input power vector using the micro-controllers and the execution time was recorded.

2. The look-up table was generated on a central processing unit (on a desktop computer) using Java.

3. The micro-controller was loaded with the look-up table (centroids/code-words and their corresponding capacity values). The input power vector was compared to the centroids in the codebook using linear and binary searches. This operation was repeated for different number of code-words for the same value of K and the corresponding time from both the search methods were recorded for each value of K and $|\mathcal{C}|$.

4. Different values of K were used.

For PIC16F877A the execution time was recorded for K=$\{1, 2, 3, \cdots, 25\}$ (K is the number of transmitter-receiver pairs/dimensions) and $|\mathcal{C}| = \{8, 16, 24\}$, (i.e., different code-

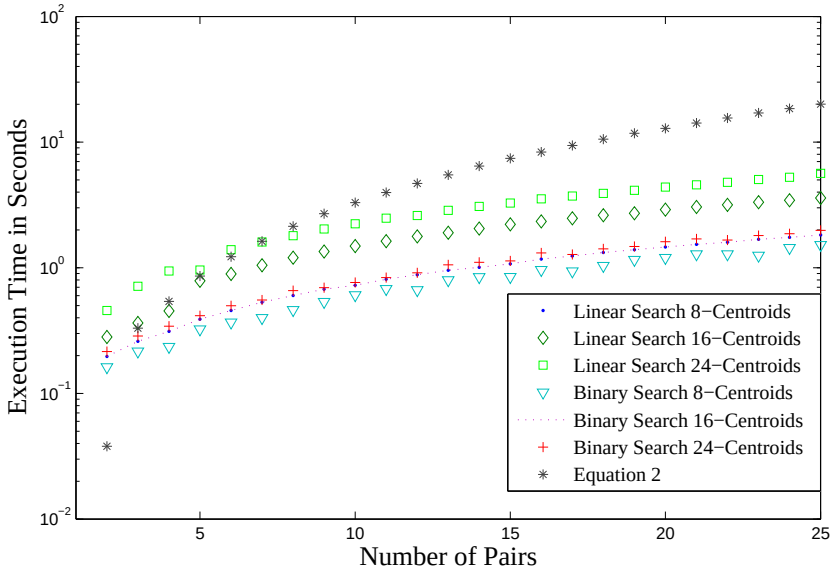


Figure 6.27 Time for computing Eqn. 4.33 vs. look-up time for PIC16F877A

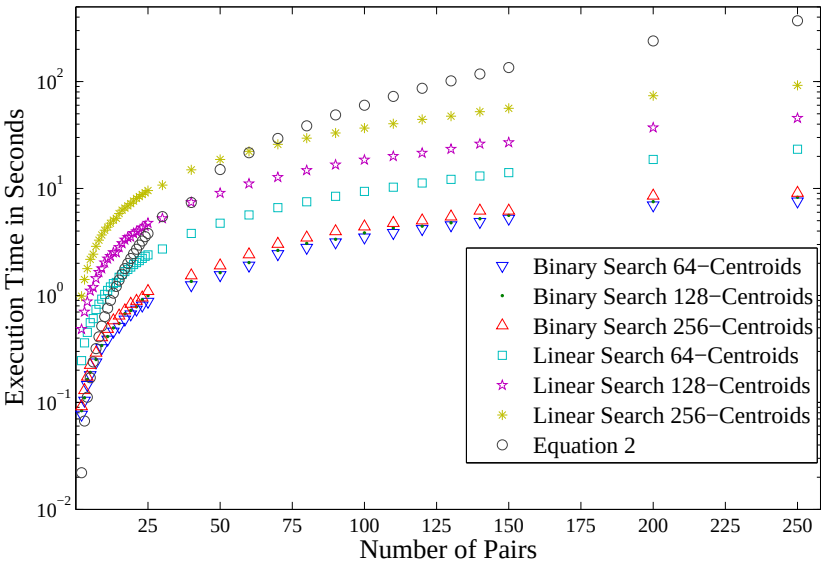


Figure 6.28 Time for computing Eqn. 4.33 vs. look-up time for PIC18F87J60

book sizes). The results are shown in Fig. 6.27. It can be seen that as K increases the time needed to compute the capacity equation grows almost exponential.

As expected, the time needed for linear search scaled linearly as the number of centroids increased. This is because the number of computations increased linearly with the number of centroids i.e., the time needed for computing 16 centroids is twice the time needed for the 8 centroids. For the 24 centroids, it is 3 times the time of the 8 centroids.

For the binary search, each split (a split increases the number of current centroids by 2) costs only one more look up which is just two more computations. For example, the 8 centroids require $2\log_2(8)$ comparisons which is 6 while the 16 centroids require $2\log_2(16)$ comparisons which is 8. For 24 centroids only 10 comparisons are needed. That is why the time difference between the binary search of $8, 16$, and 24 has close values. Interestingly, the number of comparisons for 16 centroids under binary search is 8 which is the same number of comparisons being performed in linear search with 8 centroids. This is why their execution time matches.

For PIC18F87J60, the execution time was recorded for K=$\{1, 2, \cdots, 250\}$ and $|\mathcal{C}|$=$\{1$, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024$\}$. The result is shown in Fig. 6.28. It can be seen that the results follow the same pattern which was obtained for the PIC16F877A (regarding the behavior, scaling and number of comparisons). It can be seen that for large values of K e.g., 250 pairs (dimensions), the time needed to compute the equation is 370.5 seconds while the time needed to get an answer from the code-book with $|\mathcal{C}| = 256$ entries using a linear search is 92.09 seconds and only 9.87 seconds using a binary search. This shows how the VQ

can empower a relatively slow device, originally incapable of doing the task within the time window, and enable it to get the answer in a much shorter time. Table 6.2 shows the exact number of comparisons that are performed for various number of code-word sizes ($|\mathcal{C}| = N$) in the code-book. Notably, for small values of K the time needed to calculate the capacity equation is less than that required to go through a linear search, since the latter requires going through all the centroids regardless of the number of pairs. Thus for small values of K evaluating the capacity equation out-performs the linear search.

Table 6.2 Number of Comparisons for Linear search vs. Binary search

Number of Centroids (N)	Number of Comparisons using Linear Search $\mathcal{O}(N)$	Number of Comparisons using Binary Search $\log(N)$
2	2	2
4	4	4
8	8	6
16	16	8
32	32	10
64	64	12
128	128	14
256	256	16
512	512	18
1024	1024	20
2048	2048	22

As outlined in Eqn. (4.32), the distortion increases as K increases and decreases as the number of code-words increases. In Fig. 6.29, we show the distortion that results from 15 pairs and 10 pairs versus $|\mathcal{C}|$=$\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$. The power of each pair is bound between 0 and 10 Watts.

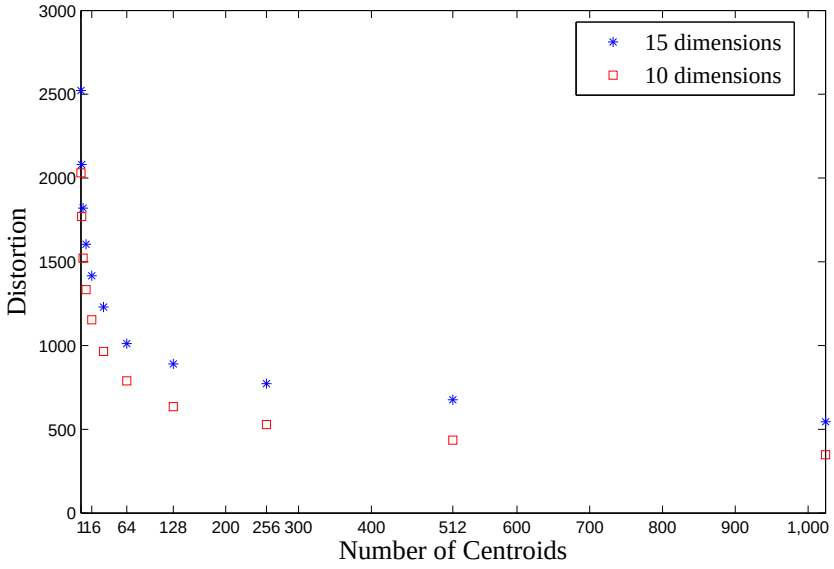


Figure 6.29 Distortion for varying number of centroids. Each dimension ranges from 0 to 10.

6.3 Performance Evaluation of Implemented DSA System

To demonstrate the efficiency of the implemented database-assisted DSA system, we compare it to a legacy system and show the benefits in terms of resilience to jamming, channel relinquishment on primary arrival, and best channel determination and allocation. Finally, we show the performance gains in terms of frame error rate.

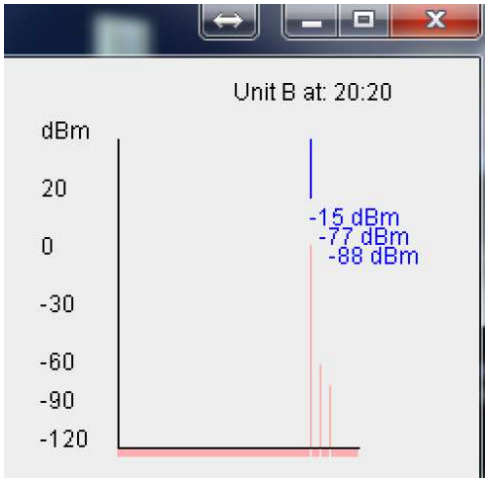


Figure 6.30 RSSI measurements at location-B for 900 MHz, 905 MHz, and 910 MHz, with the best channel being 910 MHz.

6.3.1 Deployment

Type-1 spectrum sensors were deployed in an indoor environment (laboratory) and programmed to sense 300 KHz of bandwidth centered at 900 MHz, 905 MHz, and 910 MHz. The sensed RSSI were streamed to the DSA server.

Fig. 6.30 shows an instantaneous RSSI in dBm for 900 MHz, 905 MHz and 910 MHz at a particular location (namely, location-B). During that time, any request for a channel in the vicinity of location-B would be recommended to use the 910 MHz channel since it has the lowest RSSI (i.e., -80 dBm) among the 3 channels. To induce controlled interference and mimic the primary users, we built two software configurable jammers using RFM22 transceivers as the front-end and ATmega328p as the controller. We were able to control the frequency, bandwidth, transmission power and transmit pattern of the jammers.

6.3.2 DSA versus Legacy Tx-Rx

We built two transmitter-receiver pairs with identical components (i,e., same RF front-end and controller sections). One pair was enabled with DSA capability– the pair was programmed to contact the service engine (over Ethernet) and utilize the 'channel allocation service'. The other transmitter-receiver did not have this DSA capability and worked just like legacy radios. For fair comparison, both pairs were placed equi-distant from the jammers. The Tx-Rx distance for both pairs were also the same as shown inFig. 6.31.

On start-up, both transmitters were programmed to transmit packets of 16 bits on the default channel of 900 MHz using FSK with 2 dBm of transmission power. Each packet was identified with a `sequence number`. By default, the receivers of both pairs would listen on the 900 MHz channel. On reception of a packet, the receivers would send an acknowledge (ACK) to the respective transmitter. Both the transmitter and receiver keep track of the number of packets that are successful.

The DSA pair (both transmitter and receiver) continuously i) contacts the service engine to check for primary's presence (similar to the SenseLess framework [113]), and ii) requests for a new channel if there are 5 consecutive failures[2]. On the other hand, the legacy pair continues to use the 900 MHz channel regardless of packets being successful or not. The receivers of both pairs were interfaced to a Windows machine that used a Java snippet to monitor the receivers' throughput and packer error rate.

[2]We chose 5 randomly; although in real scenarios it will depend on the type of data, mission criticality and other parameters.

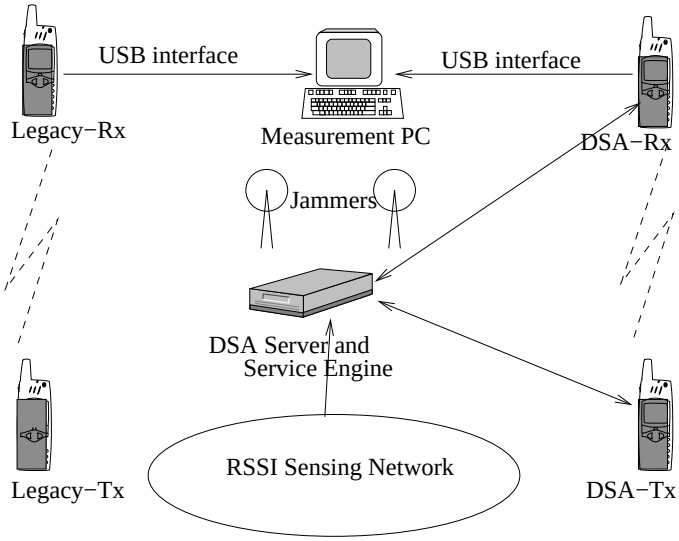


Figure 6.31 Experimental setup for DSA and legacy comparison.

6.3.3 Frame Error Rate (FER)

In addition to displaying the throughput and the packet error rate of each receiver, the Java snippet also displays additional information as shown in Figs. 6.32(a) and 6.32(b). With time on the x-axis, both figures show which channel is being used. Fig. 6.32(a) shows that channel 900 MHz is being used followed by 905 MHz, 910 MHz and again 900 MHz. Every time, the channel was changed when the 'failure' remained for more than 5 packets (also shown on the figure). On the other hand, the legacy pair continues to use the 900 MHz channel for the entire duration in-spite of the long period of 'failure' as shown in Fig. 6.32(b).

We define a *frame* as a group of 10 packets. Both Figs. 6.32(a) and 6.32(b) show small ticks on top of the success field that represent the frame times. We define frame error rate (FER) as the fraction of packets that are unsuccessful in a frame. For example, if 3 packets are unsuccessful in a frame, the FER is 0.3. For the DSA pair, it can be noted from

Fig. 6.32(a) that the FER is 0 for most frames, and either 0.1 or 0.2 for few (due to sporadic noise). However, when channel switching occurs due to the channel being jammed, it can be seen that the FER is considerably high for a small period of time. When the DSA pair switch their communication to a new channel, the FER recovers. For the legacy pair, the FER remains very high for the entire duration the channel is jammed.

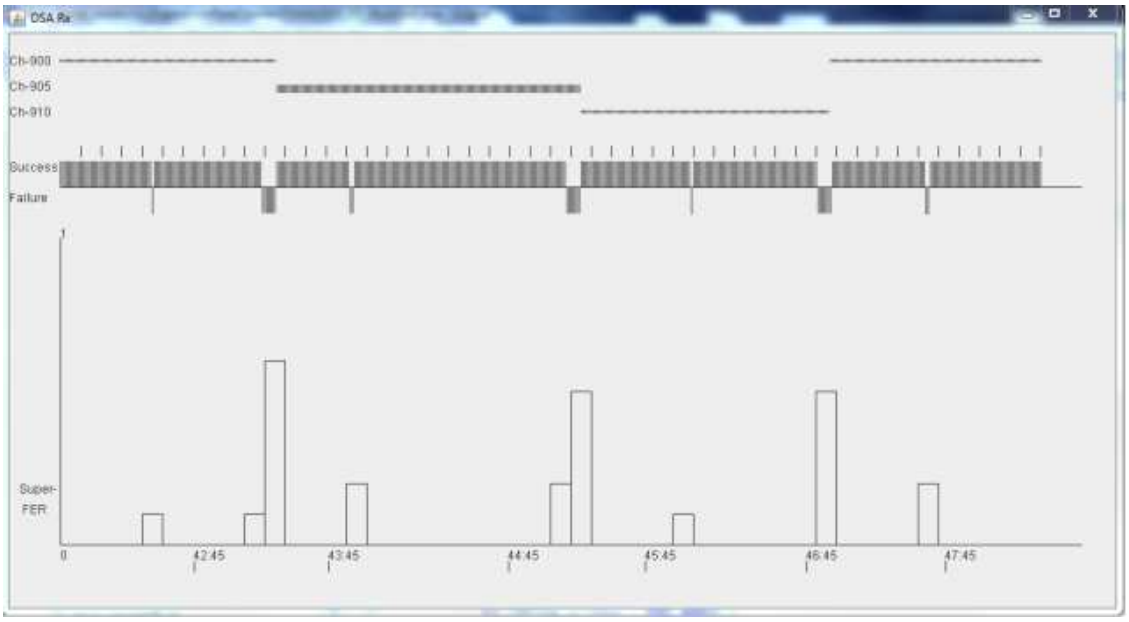
6.3.4 Jammer in Operation

Figs. 6.32(a) and 6.32(b) show that both pairs start on 900 MHz channel at t=0. Till t=43:00, packets were received successfully at both receivers with some sporadic failures due to the background noise. Jammer-1 was turned on 900 MHz at t=43:02 as shown in Fig. 6.33(a). It has a transmit power of 20 dBm. Note how the occurrence of failures at both pairs are correlated with the time at which jammer-1 is activated on 900 MHz. For the legacy pair, from Fig. 6.32(b), it is clear that the pair is totally jammed and it suffers from total packet loss (indicated by FER=1) for the entire duration jammer-1 is active on the 900 MHz. As for the DSA pair, when jammer-1 causes 5 packet failures, the DSA pair sought a new channel and contacted the DSA server which checks the RSSI values on all the candidate channels as shown in Fig. 6.34. Based on the lowest RSSI values, the DSA pair switches to 905 MHz as it was chosen by the DSA server. To further show the DSA system in work, we activate jammer-2 on 905 MHz from t=45:15. This jamming forces the DSA pair to contact the DSA server again. As a result the pair switches to 910 MHz as it was was the only available

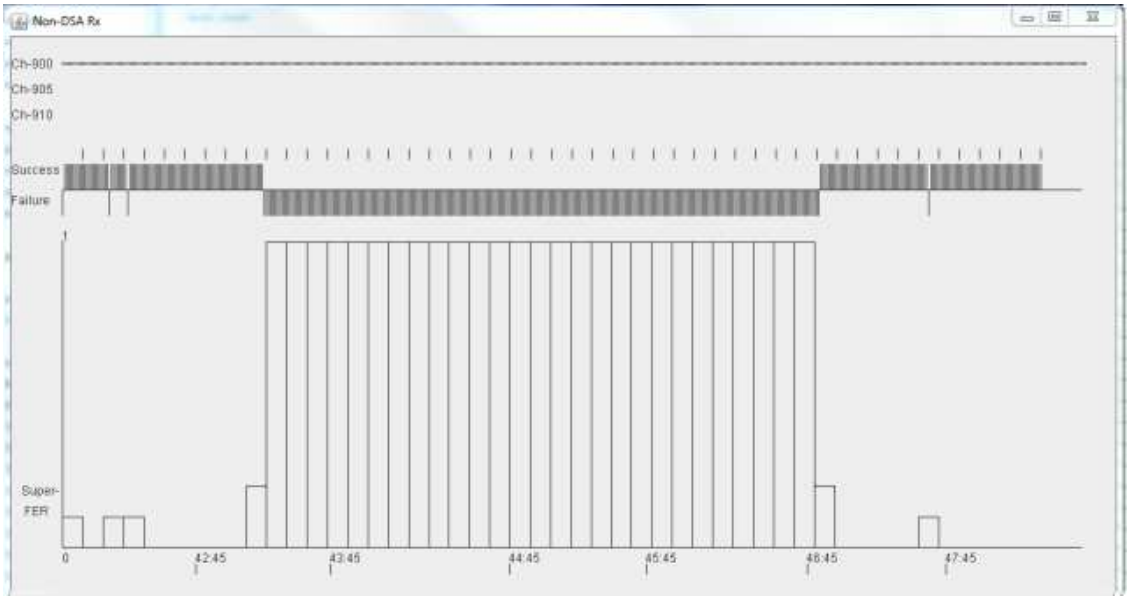
channel at t=45:018. Finally, jammer-1 switches the jamming channel from 900 MHz to 910 MHz at t=46:49. As a result, the legacy pair is able to resume successful packet reception (on 900 MHz) as indicated by Fig 6.32(b). In response to jammer-1 jamming the 910 MHz, the DSA pair again contacts the DSA server and switches from 910 MHz to 900 MHz.

6.3.5 Performance Gains

To quantify the gains obtained from the DSA pair, we compare the normalized throughput of both the pairs. The jammer emulated primary activity with a Poisson distributed 'On-Off' model with activity ranging from 0.1 to 1.0. The resultant normalized throughput values are shown in Fig. 6.35. For very low primary activity, the legacy pair outperforms the DSA pair as the latter incurs some overhead due to sensing and server access latency. When primary activity increases, the normalized throughput of the legacy pair decreased linearly. However, there was no change in the normalized throughput for the DSA pair as long as there was an available channel.

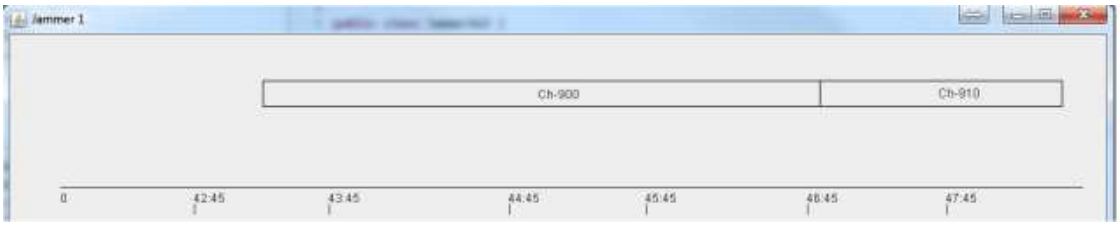


(a)

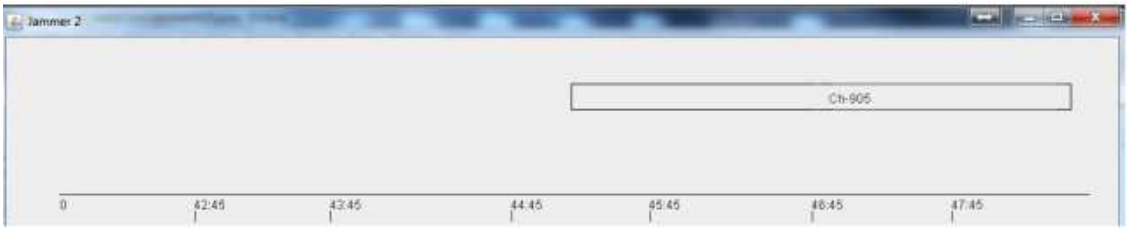


(b)

Figure 6.32 (a) Reception status of the DSA receiver. (b) Reception status of the legacy receiver.



(a)



(b)

Figure 6.33 (a) Activity of jammer-1. (b) Activity of jammer-2.

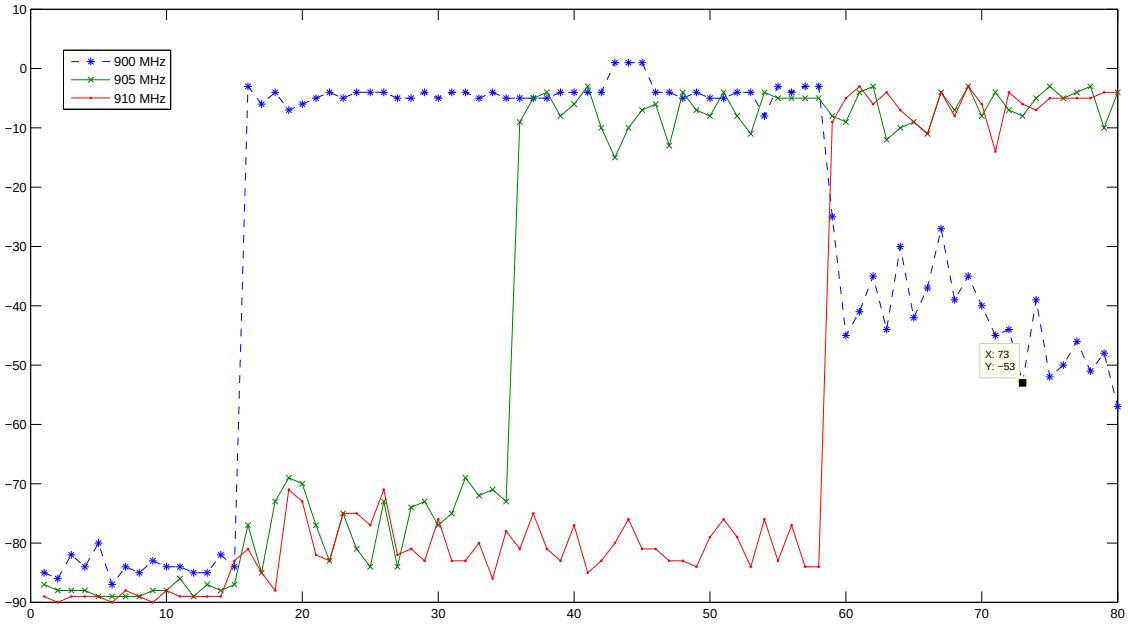


Figure 6.34 RSSI values for channels 900, 905, 910 MHz during the evaluation.

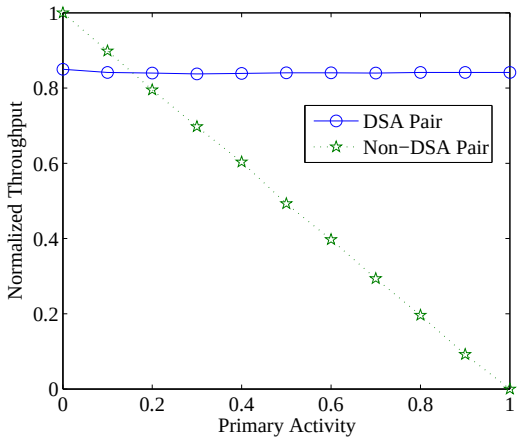


Figure 6.35 Normalized throughput of DSA and legacy pairs.

CHAPTER 7: CONCLUSIONS

This dissertation addresses some fundamental theoretical aspects on how to optimize connectivity and capacity of DSA networks, it also presents a low-cost yet effective architecture to solve the cost-scalability problem that is hindering the wide scale deployment of such networks.

With respect to connectivity, we found the conditions under which the giant component emerges in the secondary network. This is achieved by linking percolation from the Boolean model with the concept of effective density under the SINR model where we identified and excluded the nodes that are invisible to the giant component. We utilized the concept of effective density to maximize the secondary network connectivity using three different approaches. We also studied the performance of flooding and traditional flooding under the SINR model and analyzed the sources of their performance degradation. Then we employed our findings on connectivity and proposed a modified probabilistic flooding technique which outperformed the popular techniques in terms of lower message overhead and the node outreach. We also analyzed the connectivity of multi-channel distributed networks and showed the individual as well as the combined effects of the scanning limitations, interference, primary users, and the existence of multi-channels on the resultant network

connectivity. The theoretical findings were supported by the simulation experiments which showed correspondence between the theoretical and practical results.

As for the capacity, we derived the bounds on the capacity of a finite and randomly deployed secondary network in the presence of the primary users. Using dimensionality reduction, we sped up the constrained non-convex optimization problem. Then we showed how to quickly evaluate the resultant power vector using K-means clustering. We gauged the performance of our technique by running it on 8-bit microcontrollers and comparing the execution time vs. the traditional approach of evaluating a QoS metric. The results showed that our technique performed the evaluation in a shorter time.

We implemented a low-cost yet effective architecture that enables dynamic spectrum access for any type of network ranging from IoT to cellular. We showed a complete system level solution for a database-assisted DSA system that is implemented with standard servers and inexpensive software configurable RF chips, thereby achieving economics of scale. We used three main components for the system's architecture: RSSI sensing network, the DSA server and the service engine. The modular design of these components allows transparency between the components, ease of maintenance, and scalability in terms of size and features. To demonstrate the efficiency of the implemented database-assisted DSA system, we compared it to a legacy system and showed the benefits in terms of resilience to jamming, channel relinquishment on primary arrival, and best channel determination and allocation. We also showed the performance gains in terms of frame error rater and spectral efficiency.

Overall, the dissertation provides new insights on understanding and improving the connectivity and the capacity of DSA networks. It also provides low cost solutions to spectrum sensing, spectrum usage management, and value added services to the network operators. The insights from this dissertation can be used to build various algorithms, protocols, architectures and testbeds for enhanced performance and make DSA network a common technology.

LIST OF REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[2] NSF, "Spectrum occupancy measurements location 4 of 6: Republican national convention," tech. rep., National Science Foundation, September 2004.

[3] FCC, "Spectrum policy task force report, FCC 02-155," tech. rep., FCC, 2002.

[4] R. J. Berger, "Open spectrum: A path to ubiquitous connectivity.," *ACM Queue*, vol. 1, no. 3, pp. 60–68, 2003.

[5] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *Selected Areas in Communications, IEEE Journal on*, vol. 23, pp. 201–220, Feb 2005.

[6] J. Mitola, "Cognitive radio—an integrated agent architecture for software defined radio," 2000.

[7] Wireless Innovation Forum. `http://www.wirelessinnovation.org/`.

[8] International Telecommunications Union, "Essential requirements for a spectrum monitoring system for developing countries," 2011. `https://www.itu.int/dms_pubrec/itu-r/rec/sm/R-REC-SM.1392-2-201102-I!!MSW-E.docx`.

[9] M. J. Marcus, "Unlicensed cognitive sharing of tv spectrum: the controversy at the federal communications commission.," *IEEE Communications Magazine*, no. 5, pp. 1–24, 2005.

[10] S. Riess, J. Brendel, A. Stoeckle, R. Rose, and G. Fischer, "Components and implementation of a spectrum sensing network for the uhf tv band.," *2013 European Microwave Conference*, pp. 1091–1094, 2013.

[11] FCC, "Third memorandum opinion and order," 2012. `https://www.whitespacealliance.org/documents/US_TVWS_Rules_FCC-12-36A1.pdf`.

[12] Microsoft, "Microsoft spectrum observatory download and setup manual," 2014. `https://observatory.microsoftspectrum.com/SpectrumObservatoryRegistration/DownloadSetupManual`.

[13] O. Dousse, M. Franceschetti, N. Macris, R. Meester, and P. Thiran, "Percolation in the signal to interference ratio graph.," *Journal of Applied Probability*, no. 2, p. 552, 2006.

[14] P. Gupta and P. R. Kumar, "Critical power for asymptotic connectivity in wireless networks," in *Critical Power for Asymptotic Connectivity in Wireless Networks*, pp. 547–566, 1998.

[15] M. Penrose, "On k-connectivity for a geometric random graph.," *Random Structures & Algorithms*, vol. 15, no. 2, pp. 145 – 164, n.d.

[16] L. Fu, Z. Liu, D. Nie, and X. Wang, "K-connectivity of cognitive radio networks," in *Communications (ICC), IEEE International Conference on*, pp. 83–87, June 2012.

[17] L. Fu, L. Qian, X. Tian, H. Tang, N. Liu, G. Zhang, and X. Wang, "Percolation degree of secondary users in cognitive networks," *Selected Areas in Communications, IEEE Journal on*, vol. 30, pp. 1994–2005, November 2012.

[18] D. Lu, X. Huang, P. Li, and J. Fan, "Connectivity of large-scale cognitive radio ad hoc networks," in *INFOCOM, Proceedings IEEE*, pp. 1260–1268, March 2012.

[19] W. Ren, Q. Zhao, and A. Swami, "Connectivity of heterogeneous wireless networks," *Information Theory, IEEE Transactions on*, vol. 57, pp. 4315–4332, July 2011.

[20] E. N. Gilbert, "Random plane networks.," *Journal of the Society for Industrial and Applied Mathematics*, no. 4, p. 533, 1961.

[21] O. Dousse, F. Baccelli, and P. Thiran, "Impact of interferences on connectivity in ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 13, pp. 425–436, Apr. 2005.

[22] R. Meester and R. Roy, *Continuum percolation.* Cambridge tracts in mathematics: 119, New York : Cambridge University Press., 1996.

[23] R. Vaze, "Percolation and connectivity on the signal to interference ratio graph," in *INFOCOM, Proceedings IEEE*, pp. 513–521, March 2012.

[24] L. Sun and W. Wang, "Understanding topology dynamics in large-scale cognitive radio networks under generic failures," in *Global Communications Conference (GLOBECOM), IEEE*, pp. 1161–1166, Dec 2013.

[25] W. C. Ao, S.-M. Cheng, and K.-C. Chen, "Connectivity of multiple cooperative cognitive radio ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 30, pp. 263–270, February 2012.

[26] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, pp. 153–181, Kluwer Academic Publishers, 1996.

[27] G. Vakulya and G. Simon, "Percolation driven flooding for energy efficient routing in dense sensor networks," *Journal of Telecommunications and Information Technology*, pp. 103–110, 2010.

[28] V. Dimakopoulos and E. Pitoura, "On the performance of flooding-based resource discovery," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 17, pp. 1242–1252, Nov 2006.

[29] S. R. D. C. E. Perkins, E. B. Royer, "Ad Hoc On-Demand Distance Vector (AODV)," RFC 3561, RFC Editor, 2003.

[30] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proceedings of the 16th International Conference on Supercomputing*, ICS '02, (New York, NY, USA), pp. 84–95, ACM, 2002.

[31] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 479–491, June 2006.

[32] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, "A local search mechanism for peer-to-peer networks," in *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, (New York, NY, USA), pp. 300–307, ACM, 2002.

[33] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," in *Wireless Communications and Networking. WCNC. IEEE*, vol. 2, pp. 1124–1130 vol.2, March 2003.

[34] O. Dousse, P. Mannersalo, and P. Thiran, "Latency of wireless sensor networks with uncoordinated power saving mechanisms," in *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '04, (New York, NY, USA), pp. 109–120, ACM, 2004.

[35] K. Oikonomou and I. Stavrakakis, "Performance analysis of probabilistic flooding using random graphs," in *World of Wireless, Mobile and Multimedia Networks. WoWMoM. IEEE International Symposium on a*, pp. 1–6, June 2007.

[36] Y. Ma, Y. Guo, K. Niu, and J. Lin, "Transmission capacity of secondary networks in hybrid overlaid/underlaid cognitive radio systems," in *Communication Technology (ICCT), IEEE 14th International Conference on*, pp. 397–401, Nov 2012.

[37] S. Stotas and A. Nallanathan, "Enhancing the capacity of spectrum sharing cognitive radio networks," *Vehicular Technology, IEEE Transactions on*, vol. 60, pp. 3768–3779, Oct 2011.

[38] Y. Guo, Y. Ma, K. Niu, and J. Lin, "Transport capacity of cognitive radio ad hoc networks with primary outage constraint," in *Wireless Communications and Networking Conference (WCNC), IEEE*, pp. 3387–3391, April 2013.

[39] Z. Shu, Y. Yang, Y. Qian, and R. Hu, "Impact of interference on secrecy capacity in a cognitive radio network," in *Global Telecommunications Conference (GLOBECOM), IEEE*, pp. 1–6, Dec 2011.

[40] Y. Shi, C. Jiang, Y. Hou, and S. Kompella, "On capacity scaling law of cognitive radio ad hoc networks," in *Computer Communications and Networks (ICCCN), Proceedings of 20th International Conference on*, pp. 1–8, July 2011.

[41] Y. Li, X. Wang, X. Tian, and X. Liu, "Scaling laws for cognitive radio network with heterogeneous mobile secondary users," in *INFOCOM, Proceedings IEEE*, pp. 46–54, March 2012.

[42] L. Sun and W. Wang, "On study of achievable capacity with hybrid relay in cognitive radio networks," in *Global Telecommunications Conference. GLOBECOM. IEEE*, pp. 1–6, Nov 2009.

[43] S. Lirio Castellanos-Lopez, F. Cruz-Perez, M. Rivero-Angeles, and G. Hernandez-Valdez, "Voip erlang capacity in coordinated cognitive radio networks," in *Vehicular Technology Conference (VTC Fall), IEEE 78th*, pp. 1–6, Sept 2013.

[44] S. Castellanos-Lopez, F. A. Cruz-Perez, M. E. Rivero-Angeles, and G. Hernandez-Valdez, "Erlang capacity in coordinated cognitive radio networks with stringent-delay applications," in *Personal Indoor and Mobile Radio Communications (PIMRC), IEEE 24th International Symposium on*, pp. 3166–3170, Sept 2013.

[45] Y. Guo, Y. Ma, K. Niu, and J. Lin, "Transmission capacity of cognitive radio networks with interference avoidance," in *Vehicular Technology Conference (VTC Spring), IEEE 77th*, pp. 1–5, June 2013.

[46] X. Hong, C.-X. Wang, M. Uysal, X. Ge, and S. Ouyang, "Capacity of hybrid cognitive radio networks with distributed vaas," *Vehicular Technology, IEEE Transactions on*, vol. 59, pp. 3510–3523, Sept 2010.

[47] S. Gunawardena and W. Zhuang, "Capacity analysis and call admission control in distributed cognitive radio networks," *Wireless Communications, IEEE Transactions on*, vol. 10, pp. 3110–3120, September 2011.

[48] A. T. Hoang, Y.-C. Liang, and M. Islam, "Power control and channel allocation in cognitive radio networks with primary users' cooperation," *Mobile Computing, IEEE Transactions on*, vol. 9, pp. 348–360, March 2010.

[49] H.-P. Lin, K.-H. Liu, and H.-Y. Hsieh, "Design of power control protocols for spectrum sharing in cognitive radio networks: A game-theoretic perspective," in *Communications (ICC), IEEE International Conference on*, pp. 1–6, May 2010.

[50] W. Wang, T. Peng, and W. Wang, "Optimal power control under interference temperature constraints in cognitive radio network," in *Wireless Communications and Networking Conference. WCNC. IEEE*, pp. 116–120, March 2007.

[51] P. Zhou, Y. Chang, and J. Copeland, "Learning through reinforcement for repeated power control game in cognitive radio networks," in *Global Telecommunications Conference (GLOBECOM), IEEE*, pp. 1–6, Dec 2010.

[52] T. T. Nguyen, T. M. Nguyen, H. V. Nguyen, and K. L. Dang, "Hardware implementation of reception diversity techniques for spectrum sensing efficiency enhancement in cognitive radio network.," *2013 Third World Congress on Information Communication Technologies (WICT 2013)*, p. 69, 2013.

[53] T. T. Nguyen, K. L. Dang, H. V. Nguyen, and P. H. Nguyen, "A real-time fpga implementation of spectrum sensing applying for dvb-t primary signal.," *2013 International Conference on Advanced Technologies for Communications (ATC 2013)*, p. 164, 2013.

[54] y. Tachwali, Y.1, f. Basma, F.1, and h. Refai, H.1, "Adaptability and configurability in cognitive radio design on small form factor software radio platform.," *Wireless Personal Communications*, vol. 62, no. 1, pp. 1 – 29, 2012.

[55] Z. Wen, Z. Meng, Q. Wang, L. Liu, J. Zou, and L. Wang, "An fpga real-time spectrum sensing for cognitive radio in very high throughput wlan.," *Pervasive Computing the Networked World*, p. 606, 2013.

[56] T. Ha?nninen, J. Vartiainen, M. Juntti, and M. Raustia, "Implementation of spectrum sensing on wireless open-access research platform.," *2010 3rd International Symposium on Applied Sciences in Biomedical Communication Technologies (ISABEL)*, p. 1, 2010.

[57] S. Riess, J. Brendel, A. Stoeckle, R. Rose, and G. Fischer, "Components and implementation of a spectrum sensing network for the uhf tv band," in *Microwave Conference (EuMC), European*, pp. 1091–1094, Oct 2013.

[58] O. Al-Tameemi and M. Chatterjee, "Percolation in multi-channel secondary cognitive radio networks under the sinr model," in *Dynamic Spectrum Access Networks (DySPAN), IEEE International Symposium on*, pp. 170–181, April 2014.

[59] O. Al Tameemi, M. Chatterjee, and A. Al-Rumaithi, "Percolation condition for interference-limited cognitive radio networks," in *Personal Indoor and Mobile Radio Communications (PIMRC), IEEE 24th International Symposium on*, pp. 1–6, Sept 2014.

[60] O. Al Tameemi, M. Chatterjee, K. Kwiat, and C. Kamhoua, "Napf: Percolation driven probabilistic flooding for interference limited cognitive radio networks," in *Communications (ICC), IEEE International Conference on*, pp. 1–6, June 2015.

[61] O. Al-Tameemi, M. Chatterjee, K. Kwiat, and C. Kamhoua, "Distributed mac for connectivity maximization of interference limited un-coordinated dsa networks.," *MILCOM 2015 - 2015 IEEE Military Communications Conference*, pp. 103–108, 2015.

[62] O. Al Tameemi, A. Al-Rumaithi, M. Chatterjee, K. Kwiat, and K. Kamhoua, "Connectivity and rendezvous in distributed dsa networks," in *IEEE ICNC*, pp. 1–7, Feb 2016.

[63] D. Stauffer and A. Aharony, *Introduction to percolation theory.* London ; Bristol, PA : Taylor & Francis., 1994.

[64] K. Christensen, "Percolation theory." Imperial College London, 2002.

[65] P. Neelakantan and A. Babu, "Computation of minimum transmit power for network connectivity in vehicular ad hoc networks formed by vehicles with random communication range.," *International Journal of Communication Systems*, vol. 27, no. 6, pp. 931 – 955, n.d.

[66] M. Sahimi, *Applications of percolation theory.* London , Bristol, PA :Taylor & Francis., 1994.

[67] G. Grimmett, *Percolation.* New York : Springer-Verlag., 1989.

[68] B. Bollobs and O. Riordan, *Percolation.* Cambridge : Cambridge University Press., 2006.

[69] Y. Guo, Y. Ma, K. Niu, and J. Lin, "Connectivity of interference limited cognitive radio networks," in *Network Infrastructure and Digital Content (IC-NIDC), 3rd IEEE International Conference on*, pp. 158–163, Sept 2012.

[70] L. Sun and W. Wang, "Understanding blackholes in large-scale cognitive radio networks under generic failures," in *INFOCOM, 2013 Proceedings IEEE*, pp. 728–736, April 2013.

[71] D. L. Kreher and D. R. Stinson, *Combinatorial algorithms : generation, enumeration, and search.* CRC Press series on discrete mathematics and its applications, Boca Raton, Fla. : CRC Press., 1999.

[72] J. Deng, Y. S. Han, W. Heinzelman, and P. Varshney, "Balanced-energy sleep scheduling scheme for high density cluster-based sensor networks," in *4th Workshop on Applications and Services in Wireless Networks*, pp. 99–108, Aug 2004.

[73] M. Sharma and A. Sahoo, "Stochastic model based opportunistic channel access in dynamic spectrum access networks," *Mobile Computing, IEEE Transactions on*, vol. 13, pp. 1625–1639, July 2014.

[74] L. Zhai, "Opportunistic spectrum access for tdma-based cognitive radio networks.," *Optik: International Journal for Light and Electron Optics*, no. 9, p. 2081, 2014.

[75] W. Zame, J. Xu, and M. van der Schaar, "Cooperative multi-agent learning and coordination for cognitive radio networks," *Selected Areas in Communications, IEEE Journal on*, vol. 32, pp. 464–477, March 2014.

[76] A. De Domenico, E. Strinati, and M. Di Benedetto, "A survey on mac strategies for cognitive radio networks," *Communications Surveys Tutorials, IEEE*, vol. 14, pp. 21–44, First 2012.

[77] J. F. C. Kingman, *Poisson processes.* Oxford studies in probability: 3, Oxford : Clarendon Press ; New York : Oxford University Press., 1993.

[78] IEEE, "Ieee p802.22/d0.5, ieee draft standard for wireless regional area networks part 22: Cognitive wireless ran medium access control (mac) and physical layer (phy) specifications: Policies and procedures for operation in the tv bands," tech. rep., Mar 2008.

[79] S. Anand, K. Hong, S. Sengupta, and R. Chandramouli, "Is channel fragmentation/bonding in ieee 802.22 networks secure?," in *Communications (ICC), IEEE International Conference on*, pp. 1–5, June 2011.

[80] L. Jiao, I. Balapuwaduge, F. Li, and V. Pla, "On the performance of channel assembling and fragmentation in cognitive radio networks," *Wireless Communications, IEEE Transactions on*, vol. 13, pp. 5661–5675, Oct 2014.

[81] R. Irwin, A. MacKenzie, and L. DaSilva, "Resource-minimized channel assignment for multi-transceiver cognitive radio networks," *Selected Areas in Communications, IEEE Journal on*, vol. 31, pp. 442–450, March 2013.

[82] L. Yu, H. Liu, Y.-W. Leung, X. Chu, and Z. Lin, "Multiple radios for effective rendezvous in cognitive radio networks," in *Communications (ICC), IEEE International Conference on*, pp. 2857–2862, June 2013.

[83] T. Shu and M. Krunz, "Throughput-efficient sequential channel sensing and probing in cognitive radio networks under sensing errors," in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, MobiCom '09, (New York, NY, USA), pp. 37–48, ACM, 2009.

[84] S. Li, Z. Zheng, E. Ekici, and N. Shroff, "Maximizing system throughput using cooperative sensing in multi-channel cognitive radio networks," in *Decision and Control (CDC), IEEE 51st Annual Conference on*, pp. 5858–5863, Dec 2012.

[85] K. Bian and J.-M. Park, "Asynchronous channel hopping for establishing rendezvous in cognitive radio networks," in *INFOCOM, Proceedings IEEE*, pp. 236–240, April 2011.

[86] N. Theis, R. Thomas, and L. DaSilva, "Rendezvous for cognitive radios," *Mobile Computing, IEEE Transactions on*, vol. 10, pp. 216–227, Feb 2011.

[87] O. Al Tameemi, M. Chatterjee, and K. Kwiat, "Vector quantization based qos evaluation in cognitive radio networks," in *Wireless and Optical Communication Conference (WOCC), 23rd*, pp. 1–6, May 2014.

[88] O. Al Tameemi, M. Chatterjee, K. Kwiat, and C. Kamhoua, "Capacity bounds of finite secondary cognitive radio networks," in *International Conference on Computing Networking and Communications (ICNC)*, pp. 476–481, February 2015.

[89] O. Al-Tameemi, M. Chatterjee, and K. Kwiat, "Vector quantization based qos evaluation in cognitive radio networks.," *Wireless Networks*, no. 6, p. 1899, 2015.

[90] W. Guo and X. Huang, "Maximizing throughput for overlaid cognitive radio networks," in *Military Communications Conference. MILCOM. IEEE*, pp. 1–7, Oct 2009.

[91] A. Mukherjee and A. Swindlehurst, "Prescient precoding in heterogeneous dsa networks with both underlay and interweave mimo cognitive radios," *Wireless Communications, IEEE Transactions on*, vol. 12, pp. 2252–2260, May 2013.

[92] NSF, "Final report from the nsf workshop on future directions in wireless networking," tech. rep., National Science Foundation, November 2013.

[93] D.-T. Lee, "On k-nearest neighbor voronoi diagrams in the plane," *Computers, IEEE Transactions on*, vol. C-31, pp. 478–487, June 1982.

[94] R. Gray, "Vector quantization," *ASSP Magazine, IEEE*, vol. 1, pp. 4–29, April 1984.

[95] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.

[96] T. Villmann, F.-M. Schleif, M. Kaden, and M. Lange, *Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of the 10th International Workshop, WSOM 2014, Mittweida, Germany*. Springer Publishing Company, Incorporated, 2014.

[97] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *Communications, IEEE Transactions on*, vol. 28, pp. 84–95, Jan 1980.

[98] B. Huang and L. Xie, "An improved lbg algorithm for image vector quantization," in *Computer Science and Information Technology (ICCSIT), 3rd IEEE International Conference on*, vol. 6, pp. 467–471, July 2010.

[99] G. Patané and M. Russo, "The enhanced lbg algorithm," *Neural Netw.*, vol. 14, pp. 1219–1237, Nov. 2001.

[100] P. Veprek and A. Bradley, "An improved algorithm for vector quantizer design," *Signal Processing Letters, IEEE*, vol. 7, pp. 250–252, Sept 2000.

[101] E. Axell, G. Leus, E. Larsson, and H. Poor, "State-of-the-art and recent advances spectrum sensing for cognitive radio state-of-the-art and recent advances.," *IEEE Signal Processing Magazine*, vol. 29, no. 3, pp. 101 – 116, 2012.

[102] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey.," *Computer Networks*, vol. 50, pp. 2127 – 2159, 2006.

[103] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications.," *IEEE Communications Surveys and Tutorials*, vol. 11, no. 1, pp. 116 – 130, 2009.

[104] T. T. Nguyen, T. M. Nguyen, H. V. Nguyen, and K. L. Dang, "Hardware implementation of reception diversity techniques for spectrum sensing efficiency enhancement in cognitive radio network.," *2013 Third World Congress on Information Communication Technologies (WICT 2013)*, p. 69, 2013.

[105] Texas Instruments, "Overview for proprietary 2.4 ghz," 2016. `http://www.ti.com/lsds/ti/wireless_connectivity/proprietary_2-4_ghz/overview.page`.

[106] HOPERF, "RF Transceivers." `http://www.hoperf.com/rf_transceiver/`, 2016.

[107] J. Lehtomaki, J. Vartiainen, M. Juntti, and H. Saarnisaari, "Spectrum sensingwith forward methods.," *MILCOM 2006 - 2006 IEEE Military Communications conference*, p. 1, 2006.

[108] J. Vartiainen, H. Sarvanko, J. Lehtomaki, M. Juntti, and M. Latva-aho, "Spectrum sensing with lad-based methods.," *2007 IEEE 18th International Symposium on Personal, Indoor Mobile Radio Communications*, p. 1, 2007.

[109] Texas Instruments, "CC2500." `http://www.ti.com/lit/ds/symlink/cc2500.pdf`, 2016.

[110] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data.," *ACM/CSC-ER: ACM Annual Conference/Annual Meeting*, p. 517, 1968.

[111] S. B. S. Debroy and M. Chatterjee, "Spectrum map: Toward predicting the spatial distribution of spectrum usage in crns.," *ICST CROWNCOM*, pp. 1–5, 2011.

[112] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[113] R. Murty, R. Chandra, T. Moscibroda, and P. Bahl, "Senseless: A database-driven white spaces network.," *2011 IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, p. 10, 2011.