Enhancing Attack Resilience in Cognitive Radio Networks

Ruiliang Chen

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Engineering

Dr. Jung-Min Park, Chair
Dr. Y. Thomas Hou
Dr. Scott F. Midkiff
Dr. Cameron D. Patterson
Dr. Edward L. Green

February 19, 2008
Blacksburg, Virginia

Keywords: Control Channel Jamming, Cognitive Radio Networks, Multiple-Rendezvous Cognitive Media Access Control, Network Security, Primary User Emulation, Spectrum Sensing Data Falsification, Transmitter Verification, Weighted Sequential Probability Ratio Test

Enhancing Attack Resilience in Cognitive Radio Networks

Ruiliang Chen

(ABSTRACT)

The tremendous success of various wireless applications operating in unlicensed bands has resulted in the overcrowding of those bands. Cognitive radio (CR) is a new technology that enables an unlicensed user to coexist with incumbent users in licensed spectrum bands without inducing interference to incumbent communications. This technology can significantly alleviate the spectrum shortage problem and improve the efficiency of spectrum utilization. Networks consisting of CR nodes (i.e., CR networks)—often called dynamic spectrum access networks or NeXt Generation (XG) communication networks—are envisioned to provide high bandwidth to mobile users via heterogeneous wireless architectures and dynamic spectrum access techniques.

In recent years, the operational aspects of CR networks have attracted great research interest. However, research on the security aspects of CR networks has been very limited. In this thesis, we discuss security issues that pose a serious threat to CR networks. Specifically, we focus on three potential attacks that can be launched at the physical or MAC layer of a CR network: primary user emulation (PUE) attack, spectrum sensing data falsification (SSDF) attack, and control channel jamming (CCJ) attack. These attacks can wreak havoc to the normal operation of CR networks. After identifying and analyzing the attacks, we discuss countermeasures. For PUE attacks, we propose a transmitter verification scheme for attack detection. The scheme utilizes the location information of transmitters together with their signal characteristics to verify licensed users and detect PUE attackers. For both SSDF attacks and CCJ attacks, we seek countermeasures for attack mitigation. In particular, we propose Weighted Sequential Probability Ratio Test (WSPRT) as a data fusion technique that is robust against SSDF attacks, and introduce a multiple-rendezvous cognitive MAC (MRCMAC) protocol that is robust against CCJ attacks. Using security analysis and extensive numerical results, we show that the proposed schemes can effectively counter the aforementioned attacks in CR networks.

Dedicated to my parents

Gaofeng Chen and Hong Wu

and my wife

Ning Zhu

Acknowledgments

I am very fortunate to have received a lot of help from many people during my Ph.D. study. Specially, I owe an enormous debt of gratitude to my advisor, Dr. Jung-Min Park. He not only generously supported my Ph.D. study, but more importantly, passed on to me his philosophy, knowledge, and wisdom that is changing and will benefit my whole life. I will never forget the numerous times he worked so carefully to read and edit my paper work word by word, letter by letter. His serious attitude toward research and optimistic attitude toward life has kept inspiring and enlightening my mind.

I feel very lucky to have an exceptional doctoral committee and would like to thank Dr. Thomas Hou, Dr. Scott Midkiff, Dr. Cameron Patterson, and Dr. Edward Green. My dissertation would have been impossible without the guidance from them. Dr. Jeffrey Reed also gave me important directions on this research. I am very grateful to their continual support and encouragement.

During my Ph.D. research, I also had close collaboration with Randy Marchany, Michael Snow, M. Tamer Refaei, Dr. Mohamed Eltoweissy, and Kaigui Bian. It has been a very pleasant experience for me. Many of their work and inputs have been crucial to my published works and this dissertation. I should also thank Ryan Thomas, since he is the very person who first introduced to me the concept of “cognitive radio”.

I would also like to thank the fellow students in my lab whom I have worked with: Animesh Patcha, Timothy McNevin, Mike Chorzempa, Mike Snow, Kaigui, Shucai Xiao, and Yanzhu Ye. On the way to get something tough to be done, it is always helpful to have smart companies like them.

My Ph.D. study would be much more difficult and less colorful without my friends at Blacksburg. My memory is still as fresh as the day when I first came to Virginia Tech, Xiaolin Cheng, Zhenquan Jia, Yanling Li, and Animesh took great trouble helping me settle down. I could neither forget the hard time when my knee was physically disabled, how Zongmiao Wu, Li Jiang, Jing Zhou, and Kaigui helped me live through. I would like to thank them as well as Yexin Zheng, Weifeng Rao, Xiaolin Yang, Guangyin (Thomas) Lei, Tao Wang, Steven and Judy Hodges, Mike Brownfield, Tim Buennemeyer, Yumin Qi, Yan Dong, Ligang Zhang, and Jinhong Kou, for making my stay at Blacksburg such a memorable journey.

Contents

List of Figures

List of Tables

Glossary of Acronyms

ACK Acknowledgment, page 12

AOA Angle Of Arrival, page 15

BS Base Station, page 3

CAF Cyclic Autocorrelation Function, page 9

CBR Constant Bit Rate, page 110

CCJ Control Channel Jamming, page 4

CPE Consumer Premise Equipment, page 3

CR Cognitive Radio, page 2

CRC Cyclic Redundancy Check, page 28

CSD Cyclic Spectrum Density, page 9

CTS Clear To Send, page 12

CV Channel Vector, page 96

DCF Distributed Coordination Function, page 95

DDT Distance Difference Test, page 33

DRT Distance Ratio Test, page 33

DSA Dynamic Spectrum Access, page 2

DSR Dynamic Source Routing, page 110

DSS Distributed Spectrum Sensing, page 7

DSSS Direct Sequence Spread Spectrum, page 14

EIRP Equivalent Isotropically Radiated Power, page 20

FCC Federal Communications Commission, page 1

FHSS Frequency Hopping Spread Spectrum, page 14

GPS Global Positioning System, page 15

GSM Global System for Mobile communications, page 14

LRT Likelihood Ratio Test, page 78

LV Location Verifier, page 33

MAC Media Access Control, page 3

MRCMAC Multiple-Rendezvous Cognitive Media Access Control, page 4

NAV Network Allocation Vector, page 28

ND Neighbor Discovery, page 98

OSS Opportunistic Spectrum Sharing, page 2

PCBD Periodic Channel Bonding Disabling, page 102

PSTL Primary Signal Transmitter Localization, page 48

PUE Primary User Emulation, page 4

QoS Quality of Service, page 3

REM Radio Environment Map, page 96

RF Radio Frequency, page 127

RSS Received Signal Strength, page 15

RTS Request To Send, page 12

SDCCH Standalone Dedicated Control Channel, page 14

SDR Software-Defined Radio, page 2

SNR Signal-to-Noise Ratio, page 8

SPRT Sequential Probability Ratio Test, page 70

SSDF Spectrum Sensing Data Falsification, page 4

TCP Transmission Control Protocol, page 110

TDOA Time Difference Of Arrival, page 15

TOA Time Of Arrival, page 15

UDP User Datagram Protocol, page 110

WRAN Wireless Regional Area Network, page 3

WSN Wireless Sensor Network, page 49

WSPRT Weighted Sequential Probability Ratio Test, page 4

Chapter 1

Introduction

The tremendous success of wireless applications operating in unlicensed bands has resulted in the overcrowding of these bands. Unfortunately, most of the spectrum has been already allocated for licensed use, and it is difficult for the U.S. Federal Communications Commission (FCC) to allocate bands for all of the new emerging wireless applications using the current regulatory paradigms. To alleviate the spectrum shortage problem, regulators and policy makers are working on new spectrum management strategies. Specifically, the FCC is tackling the problem in three ways: spectrum reallocation, spectrum leases, and spectrum sharing [70]. In spectrum reallocation, bandwidth from government and other long-standing users is reassigned to new wireless services, such as mobile communications. In spectrum leases, the FCC relaxes the technical and commercial limitations on existing spectrum licenses by permitting existing licensees to use their spectrum flexibly for various services or even lease their spectrum to third parties. In spectrum sharing, the FCC allocates spectrum for unlicensed or shared services. While spectrum reallocation and spectrum leases focus on improving the efficiency of spectrum usage from the perspective of licensed spectrum

management, spectrum sharing aims to better regulate unlicensed spectrum usage. Spectrum sharing, in particular, has attracted great interest from regulators, manufacturers, and researchers. The FCC is considering opening up licensed bands—such as the TV band [22]—to unlicensed operations on a non-interference basis to licensed operations. Because some licensed bands (such as TV bands) are severely underutilized, spectrum sharing in fallow sections of these licensed bands can effectively alleviate the spectrum scarcity problem without causing interference to licensed operations. In the new spectrum sharing paradigm—which is called by various names such as dynamic spectrum access (DSA), opportunistic spectrum sharing (OSS), and spectrum pooling—licensed users are referred to as primary users (or incumbents) while unlicensed users that access spectrum opportunistically are referred to as secondary users (or secondaries).

The technology of cognitive radio (CR) [25, 43, 44] plays an important role in realizing the DSA paradigm. In order to achieve the highly flexible operating characteristics required for DSA, most experts agree that CRs need to be (predominately) software-based systems, instead of being hardware-based ASIC (application specific integrated circuit) devices as in conventional radios. Henceforth, we will assume that CRs are software-defined radios (SDRs). A CR can learn from its environment and intelligently adjust its operating parameters based on what has been learned. In DSA, this means that a CR needs to learn the spectrum usage status of a band and intelligently decide whether to access the band or not. The process of learning the spectrum usage status is called spectrum sensing. The technique of spectrum sensing is crucial to the successful realization of DSA. During spectrum sensing, if a secondary user detects that it is within the primary user's protection region[1] in a particular band, it will refrain from accessing that band and search for other fallow sections of spectrum. If a secondary user detects no primary users but only the presence of other secondary users during spectrum sensing, the secondary user will coordinate with the

[1]A primary user's protection region is defined as the area in which secondary users cannot operate while the primary user is in operation so that no interference to the primary user is introduced.

other secondary users to share the spectrum resource. This coordination process is also known as spectrum sharing[2] or cognitive media access control (MAC). An ideal cognitive MAC protocol should efficiently utilize available spectrum resources while meeting other requirements, including Quality of Service (QoS) and security.

Depending on the deployment scenario, the secondary users can employ either a cellular network architecture or an ad hoc network architecture. A cellular CR network architecture is employed in the IEEE 802.22 standard, which specifies the air interface (physical and MAC layers) for a CR-based Wireless Regional Area Network (WRAN) [30]. A WRAN cell is composed of a base station (BS) and numerous Consumer Premise Equipments (CPEs), and the coverage range of the BS can range from tens of kilometers to a hundred kilometers. In the cellular architecture, the BS is a master that manages the WRAN while the CPEs are slaves that directly communicate with the BS. The target application of 802.22 is to provide wireless broadband access in rural and remote areas. In contrast, an ad hoc CR network is comprised of mobile computing devices equipped with CRs, and they interact with each other via multi-hop wireless links. The establishment of each wireless link is via DSA. While CR-based WRANs are in the process of being standardized, ad hoc CR networks have been the major focus of the research community. In this dissertation, unless stated otherwise, a CR network refers to an ad hoc CR network.

1.1 Problem Statement and Motivation

Designing appropriate spectrum sensing and cognitive MAC schemes for CRs and CR networks are very important to the realization of the DSA paradigm. There has been a large body of research

[2]Note that the term "spectrum sharing" can convey two different meanings, depending on the context in which it is used. In spectrum management, it refers to the process of allocating spectrum for unlicensed or shared services. In the context of a CR network's operational functions, it refers to the network's media access control function.

on the operational aspects of the two topics [6, 7, 24, 30, 34, 41, 42, 52, 53, 54, 58, 62, 64, 76, 81]. However, very little has been done to address security issues related to the topics. Because they also pose unique security problems, which are challenging to address, those security problems must be addressed.

1.2 Research Contribution

CR networks differ from conventional wireless networks in many aspects. The most important difference lies in the physical and the MAC layers. This research focuses on three potential attacks that can be launched at the physical layer or at the MAC layer of a CR network—primary user emulation (PUE) attacks, spectrum sensing data falsification (SSDF) attacks, and control channel jamming (CCJ) attacks. The aforementioned attacks can wreak havoc to the normal operation of CR networks. After identifying and analyzing the attacks, we discuss methods to counter them. For PUE attacks, we propose a transmitter verification scheme for attack detection. The scheme utilizes the location information of a transmitter together with its signal characteristics to verify primary users and detect PUE attackers. For both SSDF attacks and CCJ attacks, we seek countermeasures for attack prevention. In particular, we propose Weighted Sequential Probability Ratio Test (WSPRT) as a data fusion technique that is robust against SSDF attacks and propose a multiple-rendezvous cognitive MAC protocol (MRCMAC) that is robust against CCJ attacks. Using security analysis and extensive simulation results, we show that these proposed schemes can effectively counter the aforementioned attacks in CR networks.

This research represents one of the very first attempts to systematically investigate security issues at the physical and the MAC layers of CR networks. The results of this research will help to minimize potential security vulnerabilities in the design and implementation of CR networks. To

be more specific, the contribution of this research is twofold. First, we pinpoint three security threats in CR networks, i.e., PUE attacks, SSDF attacks, and CCJ attacks. These threats raise new problems either because they only exist in networks or they require different solutions because of the unique characteristics of CR networks. Recognizing and understanding the threats are the important first step toward securing a CR network. Second, for each of the identified attacks, we propose a countermeasure. By conducting either attack mitigation or attack detection, the proposed countermeasures could help enhance attack resilience of a CR network.

1.3 Document Organization

This dissertation is organized as follows. Chapter 2 gives an overview of the technical background and related work. Chapter 3 describes and analyzes the aforementioned three attacks in CR networks. From Chaper 4 to Chapter 6, we propose countermeasures against each of the three attacks and using simulation and analysis to evaluate them, respectively. In particular, Chaper 4 presents a transmitter verification scheme for detecting PUE attacks, Chaper 5 elaborates a novel data fusion technique that is robust against SSDF attacks, and Chapter 6 describes the overall framework of a new cognitive MAC protocol that is robust against CCJ Attacks. In Chapter 7, we make a summary of the thesis and discuss the future research.

Chapter 2

Background

This chapter introduces the background required for this research. It provides an overview of the operational components of CR networks where potential security problems could arise. The chapter also surveys existing research related to the potential security problems and their countermeasures.

Section 2.1 defines a CR and summarizes its characteristics. This chapter continues with Section 2.2 and Section 2.3, which discuss two important operational aspects of a CR network—spectrum sensing and MAC, respectively. Next, Section 2.4 provides an overview of jamming attacks in conventional wireless networks. This is followed by a review of existing research on the localization problem in wireless networks, which is closely related to some countermeasures to PUE attacks that will be discussed in Chapter 4. We conclude this chapter with a summary in Section 2.6.

2.1 Cognitive Radio

The concept of CR was first proposed by Joseph Mitola III [43, 44]. The FCC formally defined CR as a radio that *has the technical capability to adapt their use of the spectrum in response to information external to the radio* [21]. Such a definition implies two characteristics of a CR: cognitive capability and reconfigurability [2]. Cognitive capability refers to the ability of a CR to learn information from its radio environment. Reconfigurability refers to the ability of a CR to be dynamically programmed in response to what is learned from the radio environment. The reconfigurability is enabled by the SDR technology, which is a practical reality today [25]. In the DSA paradigm, the definition of CR specifically means that CRs used by secondary users need to be able to scan a certain spectrum range and intelligently decide which spectrum band to use for its transmission. Accordingly, the cognitive capability specifically refers to the ability to detect temporally unused spectrum, i.e., *spectrum hole* or *white space*, and the reconfigurability refers to the ability to dynamically vary the modulation scheme, transmission power, time, and frequency.

2.2 Spectrum Sensing

The cognitive capability of a CR is realized in the form of spectrum sensing. This important function helps a CR learn the spectrum holes in its radio environment. The existing spectrum sensing techniques can be divided into two categories: local spectrum sensing and distributed spectrum sensing (DSS). Next, we provide an overview of both techniques.

2.2.1 Local Spectrum Sensing

Local spectrum sensing techniques include energy detection, matched filter detection, and cyclostationary feature detection.

2.2.1.1 Energy Detection

Energy detection is the simplest technique for local spectrum sensing. An energy detector infers the existence of an primary user based on the measured signal energy level. To measure the signal energy level in a band, the received signal is first processed using a bandpass filter and then the output signal is squared and integrated over an observation interval. The output of the integrator is compared with a predefined threshold to decide whether the band is being used or not. When a receiver has no sufficient information about the primary user signal, such as the characteristics of the primary user signal or the power of the random Gaussian noise, an energy detector is optimal [62].

2.2.1.2 Matched Filter Detection

When there is *a priori* knowledge about the primary user signal, such as its modulation type, pulse shape, pilot, preambles, synchronization codes, and etc., the matched filter is the optimal detector in stationary Gaussian noise because it maximizes the received signal-to-noise ratio (SNR) [62]. The advantage of a matched filter is that it requires less number of samples compared to an energy detector. The matched filter can also potentially distinguish different signal types in a band. On the other hand, a disadvantage of the matched filter is that its performance heavily depends on the accuracy of the *a priori* knowledge about the primary user signal.

2.2.1.3 Cyclostationary Feature Detection

Modulated signals are generally coupled with sine wave carriers, thereby exhibiting periodicity in their signal structure. Cyclostationary feature detection utilizes the cyclic feature of a signal to detect it. For example, the cyclic autocorrelation function (CAF) and the cyclic spectrum density (CSD) can both be used to detect signal features [52]. The advantage of the cyclostationary feature detection includes its ability to distinguish different signal types in a band and its robustness against stationary noise with unknown variance. The disadvantage of this technique lies in its computational complexity and long observation time.

2.2.2 Distributed Spectrum Sensing

In a wireless channel, signal fading can cause the signal strength at a receiver to be significantly lower than what is predicted by path loss models. There are two types of fading: shadow fading and multipath fading. Shadow fading (a.k.a. slow fading) is frequency independent and it does not cause significant fluctuations in signal strength over small changes in receiver location, while multipath fading (a.k.a. fast fading) is frequency dependent and can vary significantly with small changes in location. The effect of fading, shadow fading in particular, can result in the "hidden node problem." The hidden node problem in the context of CR networks can be described as an instance in which a secondary user in a CR network is within the protection region of an operating primary user but fail to detect the existence of the primary user. Fig. 2.1 shows two scenarios in which the hidden node problem may occur.

Recent research results [24, 42, 52, 64] indicate that the hidden node problem can be alleviated by requiring multiple secondary users to cooperate with each other in spectrum sensing—i.e., DSS. An illustration of DSS is shown in Fig. 2.2. In DSS, each secondary user acts as a sensing

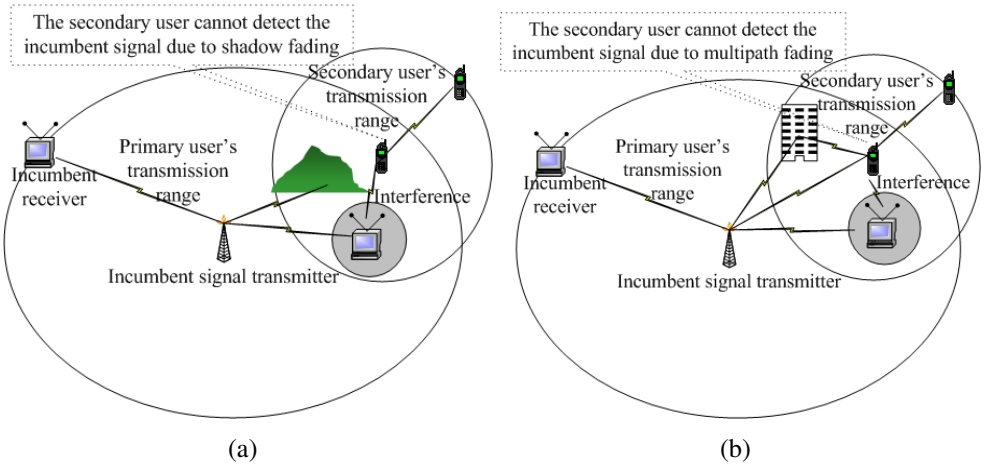


Figure 2.1: The hidden node problem caused by (a) shadow fading and (b) multipath fading.

terminal that conducts local spectrum sensing. The local results are gathered at a data collector (or "fusion center") that executes data fusion and determines the final spectrum sensing result. In an 802.22 WRAN, DSS can be implemented in a straightforward manner: the BS acts as the data collector while the CPEs serve as sensing terminals. In an ad hoc CR network, where each node is a secondary user equipped with a CR, each node acts as both a sensing terminal and a fusion center. It sends its local sensing measurements to its neighbors and executes data fusion using the measurements received from its neighbors. One advantage of DSS over non-cooperative local spectrum sensing by an individual terminal is its ability to reduce the variance of the spectrum sensing process. Furthermore, to overcome the hidden node problem with a single CR, the CR must have high enough sensitivity to detect even extremely weak primary user signals. The high cost of such highly sensitive CR terminals may limit the wide deployment of CR networks. With DSS, reliable primary user signal detection with low-cost, low sensitivity CR terminals is possible.

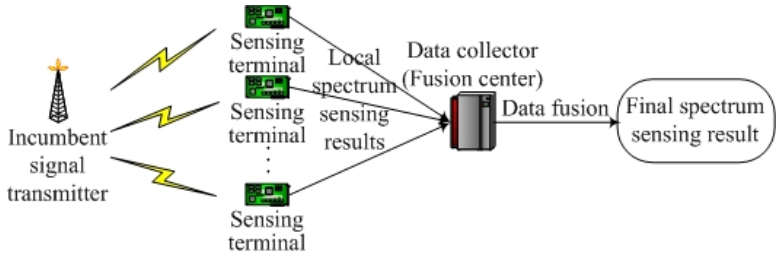


Figure 2.2: Distributed spectrum sensing.

2.3 Cognitive MAC Protocols

Once secondary users learn the available spectrum using spectrum sensing, they need to coordinate with each other to allocate the spectrum resources and dynamically change the allocation when primary users reclaim any spectrum. This is the task of a cognitive MAC protocol, which is another active research topic. Existing proposals for cognitive MAC protocols can be divided into two categories.

The first category of protocols take the centralized approach [6, 7, 30, 58, 76]. That is, a centralized entity in a CR network controls the spectrum allocation and access rules for the network. For example, the centralized entity can be physically centralized such as a BS in a 802.22 WRAN [30], and it can also be logically centralized such as a DSAP (Dynamic Spectrum Access Protocol) server [6] or a Regional Spectrum Broker [7] in an ad hoc network. The advantage of this approach is its simplicity in design and ease in achieving optimal spectrum access efficiency or fairness. However, the construction of a centralized infrastructure may not be preferable since it could become a bottleneck of a network. Therefore, the research community has been more interested in a more robust distributed approach.

In the distributed approach, secondary users build up peer-to-peer ad hoc communications with each other based on DSA. Among the distributed solutions, some MAC protocols are derived from conventional wireless MAC protocols.

For example, the DOSS (Dynamic Open Spectrum Sharing) protocol [41] is derived from MAC protocols based on busy tones. In DOSS, when spectrum opportunities are detected, a CR network sets up three operational channels for node communication: a common control channel, a busy tone channel, and a data channel. While DOSS can effectively solve the hidden node problem, the busy tone channel requires at least an additional transceiver for each node.

The schemes proposed in [32, 47, 53, 54] are all extended from the 802.11 MAC. A common idea behind these schemes is to use a common control channel to transmit Request-To-Send (RTS) and Clear-To-Send (CTS) frames and use data channels to transmit DATA and Acknowledgment (ACK) frames. However, the schemes differ in their targeting applications or specific objectives. The AS-MAC protocol [54] is proposed to coexist with a GSM cellular network. In AS-MAC, a CR network uses one of the control channels in the GSM band as the common control channel. The CR MAC [53] protocol is designed for emergency CR networks where packet transmission delay needs to be minimal. The protocol allocates a sufficiently wide control channel so that the RTS/CTS control frames will encounter minimal collisions. The scheme proposed in [47] features exchanging neighboring node information regarding channel availability, i.e., considering DSS in the cognitive MAC protocol. The HC-MAC protocol considers hardware limitation of a secondary user with a single transceiver. It focuses on the optimization of each CR's time allocation on spectrum sensing and spectrum access. Thus, HC-MAC can maximize spectrum utilization while timely responding to spectrum variability.

The HD-MAC protocol [81] is a modification of a MAC protocol designed for multi-channel 802.11 networks. HD-MAC utilizes distributed coordination to elect a control channel for each group of secondary users that are closely located. Therefore, it does not rely on a common control channel for the whole network. The protocol proposed in [15] uses a similar idea, but different from HD-MAC, which is an ad hoc Cognitive MAC protocol, the protocol in [15] is proposed for

CR-based mesh networks (i.e., CogMesh).

There are also some schemes dealing with specific CR networks. For example, the scheme proposed in [34] addresses the design of a MAC for a CR network with relatively small size and fixed number of secondary users.

In addition to the design of cognitive MAC protocols, there is also research focusing on specific topics facilitating the protocols. In particular, two research topics have attracted the most attention. One is to optimize spectrum scheduling to maximize throughput [80], enhance fairness [55, 77], or decrease communication overhead [12]. The other is to facilitate dynamic switch among different MAC protocols [18, 72].

2.4 Jamming Attacks in Wireless Networks

A jamming attack, a.k.a. a radio interference attack, is one of the simplest forms of denial-of-service attacks in a wireless network. It is an attack in which an attacker called a *jammer* purposefully tries to interfere with the physical transmission and reception of wireless communications [73]. Jamming attacks pose a serious threat to the operation of a wireless network, regardless of the network type. Existing research has investigated the effectiveness of a jamming attack in GSM networks [20, 67], wireless sensor networks (WSNs) [36, 37], wireless LANs (WLANs) [67], and ad hoc networks [50].

The primitive form of a jamming attack is to constantly interfere with physical transmission and reception. However, it has been shown that by strategically jamming control channels[1] in a wireless system, a jamming attack can be much more effective. For example, in [50], it was shown that

[1]Depending on the specific wireless system, the "control channel" here can be a frequency band, a sequence of time slots, or a collection of packets.

jamming the checksum field of 802.11 frames could make the jamming cost as low as 10^{-4} the cost for communication; some other research [20] showed that jamming the Standalone Dedicated Control Channel (SDCCH) could bring down the Global System for Mobile communications (GSM) network in Manhattan if over 110 short messages were injected into the network per second.

While jamming attacks are easy to launch and effective in wreaking havoc, defense against jamming attacks is not trivial. Conventional countermeasures have mainly focused on physical layer techniques, most notably spread spectrum techniques, including direct sequence spread spectrum (DSSS) and frequency hopping spread spectrum (FHSS) [56]. However, in practice, because of implementation difficulty, these techniques have not seen widespread deployment in today's many distributed, low-cost wireless networks, such as wireless local area networks (WLANs). Therefore, people are looking for alternative countermeasures. In [73], the authors tries to understand the modes of jamming attacks that could be launched against existing wireless networks, which could help the detection of the attacks. The research in [46] and [38] proposed channel hopping and mechanism hopping, respectively, to defend against jamming attacks. The channel hopping scheme took advantage of the fact that multiple channels exist in 802.11 networks and using channel hopping among the channels in a pseudo-random sequence could thwart jammers who have no knowledge about the sequence. The mechanism hopping scheme was based on the observation that a jamming attack has different effectiveness when launched against different protocols or protocol configurations. Also, different jamming attacks have different effectiveness when launched against the same protocol. Therefore, if one can dynamically change the protocols in any of the physical, link, network, and transport layers, the resilience against jamming attacks could be enhanced.

2.5 Localization in Wireless Networks

The localization capability may be very helpful to the countermeasures of some security threats in CR networks (see Chapter 4 for details). In this section, we summarize the existing localization techniques in wireless networks.

The conventional localization approaches are based on one or several of the following techniques: Time Of Arrival (TOA), Time Difference Of Arrival (TDOA), Angle Of Arrival (AOA), and Received Signal Strength (RSS).

The Global Positioning System (GPS) [74] is a typical localization system based on TOA. A mobile node receives signals from satellites that contain their location and time information. Based on the information, the node can calculate its own position.

TDOA is a passive localization technique that utilizes the difference between the arrival times of pulses transmitted by a transmitter but does not rely on any knowledge of the pulse transmission time. The technique measures the time differences at multiple receivers with known locations and subsequently computes a location estimate [17].

In the AOA technique, a receiver measures the angle of arrival from two or more transmitters. If the locations of the transmitters are known, the receiver can calculate its own location using triangulation [49]. Using the same principle, angle of arrival information from multiple receivers can be used to determine the transmitter's location.

RSS-based localization techniques arise from the fact that there is a strong correlation between the distance of a wireless link and RSS [39, 57]. Specifically, given a transmitter-receiver pair, RSS can be modeled as a function of transmitted power and transmitter-receiver distance. Therefore, if a correct model is used and there are multiple observers taking RSS measurements from a transmitter, then the transmitter location can be estimated using the model. For example, Wireless

E911 [23] uses "location signature" for localization, i.e., stores and matches multipath patterns (fingerprints) that mobile phone signals are known (via on-site calibration) to exhibit at different locations.

Recently there is also emerging research on the secure localization problem [9, 10]. The research aims to ensure reliable localization results. In [9] covert base stations are used when the TDOA technique is applied, making it difficult for an attacker to manipulate the timing of signal transmission or to lie about its location. In [10], a distance bounding protocol is repeatedly used to determine the upper bound distances from a device being located to multiple points with know positions. This technique can help compute a reliable location of the device and help verify the location of the device.

2.6 Chapter Summary

This chapter provided an overview of the information and literature that form the foundation for this research. Spectrum sensing and cognitive MAC protocols are two major operational aspects of CR networks that are being actively researched. Jamming attacks are a serious threat to conventional wireless networks, thus deserving more study in the context of CR networks. Localization is an important tool to assist normal operation or support security features in a wireless network.

Although spectrum sensing and cognitive MAC protocols are active areas of research, relevant security issues have yet to be studied. The security aspects of spectrum sensing and cognitive MAC protocols need to be addressed before the benefits of CR technology can be fully reaped. In the next chapter, we describe and analyze three security threats to DSS and cognitive MAC in CR networks. Identifying these threats and understanding their mechanism will serve as an important first step toward enhancing the attack resilience of CR networks.

Chapter 3

Security Threats in Cognitive Radio Networks

This chapter discusses and analyzes three security threats at the physical layer and link layer of CR networks: PUE, SSDF, and CCJ. Identifying these attacks and understanding their mechanisms compose the foremost step to develop countermeasures against them and enhance attack resilience of CR networks. Many security threats that exist in conventional wireless networks (e.g., eavesdropping, replay, spoofing) also apply to CR networks. However, this chapter only focuses on those attacks that either are unique to CR networks or require different mitigation techniques in CR networks.

This chapter is organized as follows. Section 3.1 introduces PUE attacks and discusses their disruptiveness in CR networks. Section 3.2 describes SSDF attacks and explains why existing data fusion techniques are vulnerable to the attacks. Then in Section 3.3, we cover the details about CCJ attacks in CR networks and show how existing CR MAC protocols are vulnerable to this attack. This chapter ends with a summary in Section 3.4.

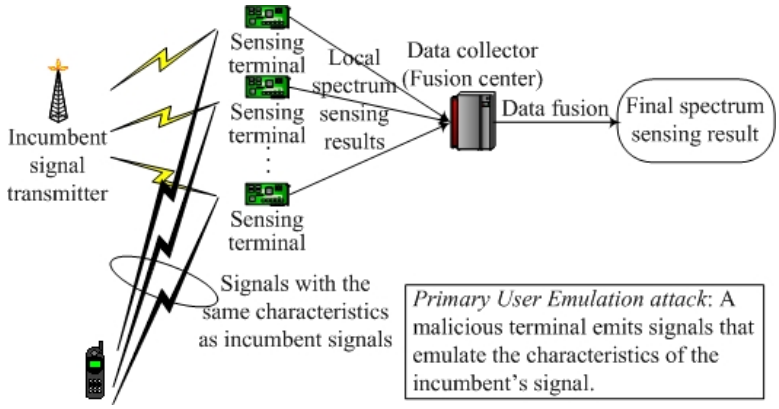


Figure 3.1: An primary user emulation attack.

3.1 Primary User Emulation Attacks

In the DSA paradigm, when a primary user is detected in a given band, all secondary users should avoid accessing that band. However, when a secondary user is detected, other secondary users may choose to share that same band. In other words, primary users have higher priority than secondary users in accessing spectrum resources. In a PUE attack, a malicious secondary tries to gain priority over other secondary users by transmitting signals that emulate the characteristics of a primary user's. An illustration of a PUE attack is shown in Fig. 3.1. Due to the reconfigurability of CRs, it is possible for an adversary to modify the radio software of a CR to change its emission characteristics (i.e., modulation, frequency, power, and etc.) so that the emission characteristics resemble those of a primary user. The potential impact of the PUE attacks depends on the legitimate secondary users' ability to distinguish the attacker's signal from actual primary signals while conducting spectrum sensing. Here we examine existing local spectrum sensing techniques and explain why they may be vulnerable to PUE attacks.

The energy detection technique (see 2.2.1.1) infers the existence of a primary user based on the measured signal energy level. Obviously, energy detection is unable to distinguish primary signals and secondary signals. An improved scheme proposed in [40] suggests the use of periodic "quiet

periods". During a quiet period, all secondary users refrain from transmitting to facilitate spectrum sensing. When quiet periods are observed by all secondary users, detecting primary users becomes straightforward—i.e., any terminal whose received signal energy level is beyond a given threshold can be considered a primary transmitter. However, such a detection strategy breaks down completely when malicious secondary users deliberately transmit during quiet periods.

Cyclostationary feature detection or matched filter detection (see 2.2.1.3 and 2.2.1.2) belongs to signal feature detection techniques, which capture special characteristics of a primary signal. However, relying solely on signal feature detection may not be sufficient to reliably distinguish primary signals from those of an attacker's. For example, in a CR network where primary users are TV systems, an attacker may emit signals that emulate TV signals. Alternatively, the attacker can replay TV signals that were previously recorded. In either case, signal feature detection will falsely identify the attacker's signal as that of a primary user.

Depending on the motivation behind the attack, a PUE attack can be classified as either a selfish PUE attack or a malicious PUE attack.

- *Selfish PUE attacks*: In this attack, an attacker's objective is to maximize its own spectrum usage. When selfish PUE attackers detect a fallow spectrum band, they prevent other secondary users from competing for that band by transmitting signals that emulate the signal characteristics of primary user signals. This attack is most likely to be carried out by two selfish secondary users whose intention is to establish a dedicated link.

- *Malicious PUE attacks*: The objective of this attack is to obstruct the DSA process of legitimate secondary users—i.e., prevent legitimate secondary users from detecting and using fallow licensed spectrum bands, causing denial of service. Unlike a selfish attacker, a malicious attacker does not necessarily use fallow spectrum bands for its own communication purposes. It is quite possible for an attacker to simultaneously obstruct the DSA process

in multiple bands by exploiting two DSA mechanisms implemented in every CR. The first mechanism requires a CR to wait for a certain amount of time before transmitting in the identified fallow band to make sure that the band is indeed unoccupied. Existing research shows that this time delay is non-negligible [13, 64]. The second mechanism requires a CR to periodically sense the current operating band to detect primary signals and to immediately switch to another band when such signals are detected. By launching an PUE attack in multiple bands in a round-robin fashion, an attacker can effectively limit the legitimate secondary users from identifying and using fallow spectrum bands.

Note that in PUE attacks, the adversary only transmits in fallow bands. Hence, interference to primary users is not a concern. We carried out simulation experiments to showcase the disruptive effects of PUE attacks. In the simulated network, 300 secondary users (which include both legitimate and malicious users) are randomly located inside a 2000m$\times$2000m square area, each with a transmission range of 250m and an interference range of 550m. These range values are consistent with the protocol interference model [31]. Two TV broadcast towers act as primary signal transmitters. Each TV tower has ten 6MHz channels, and the duty cycle of all the channels is fixed at 0.2. One tower is located 8000m east of the square area and has a transmission radius of 9000m; the other tower is located 5000m south of the square area with a transmission radius of 7000m[1]. The layout of the simulated network is shown in Fig. 3.2(a). Each secondary user node is randomly placed in the network area and moves according to a random waypoint model [4] by repeatedly executing the following four steps: 1) It randomly chooses a destination in the square area with a uniform distribution; 2) It chooses a velocity v that is uniformly distributed over [v_{min}, v_{max}]; 3) It moves along a straight line from its current position to the destination with velocity v; and 4)

[1]We set the values of 9000m and 7000m for the primary users' transmission radiuses based on realistic assumptions. Suppose the following parameters: the equivalent isotropically radiated power (EIRP) of the TV towers (transmitters) is 2500KW, transmitters' effective antenna height is 100m, receivers' effective antenna height is 1m, and receivers' energy detection sensitivity is –94dbm. Under these conditions, one can derive a transmission radius of 8000m using the rural environment version of the HATA model [57].

It pauses in the destination for a random period that is uniformly distributed over [0, t_{p-max}]. We chose the values $v_{min} = 5$m/s, $v_{max} = 10$m/s, and $t_{p-max} = 60$s. Each simulation instance spans a period of 24 hours. Another one hour before the 24 hours was simulated to ensure that the random waypoint model entered steady state. The number of attackers was varied from 1 to 30.

Figs. 3.2(b) and 3.2(c) show the simulation results for the selfish PUE attack and the malicious PUE attack, respectively. The y-axis in the figures represents the amount of link bandwidth each secondary user is able to detect. The results show that a selfish PUE attack can effectively steal bandwidth from legitimate secondary users while a malicious PUE attack can drastically decrease the link bandwidth available to legitimate secondary users.

3.2 Spectrum Sensing Data Falsification Attacks

The second security threat to DSS is the transmission of false spectrum sensing data by malicious secondary users. An attacker may send false local spectrum sensing results to a data collector, causing the data collector to make a wrong spectrum sensing decision. We use the term SSDF attack to refer to such an attack. The attack is illustrated in Fig. 3.3. To maintain an adequate level of accuracy in the midst of SSDF attacks, the data fusion technique used in DSS needs to be robust against fraudulent local spectrum sensing results reported by malicious secondary users. Although a few data fusion techniques for DSS have been proposed recently, none of them addresses this problem.

In the following, we describe three data fusion techniques that were proposed recently for DSS. We describe each technique briefly and discuss its vulnerability to SSDF attacks. To facilitate our discussion, we model the DSS process as a parallel fusion network, as shown in Fig. 3.4. In this figure, N_0 is a data collector, N_i ($i = 0, 1, 2, \ldots, m$, where m is the number of N_0's neighboring

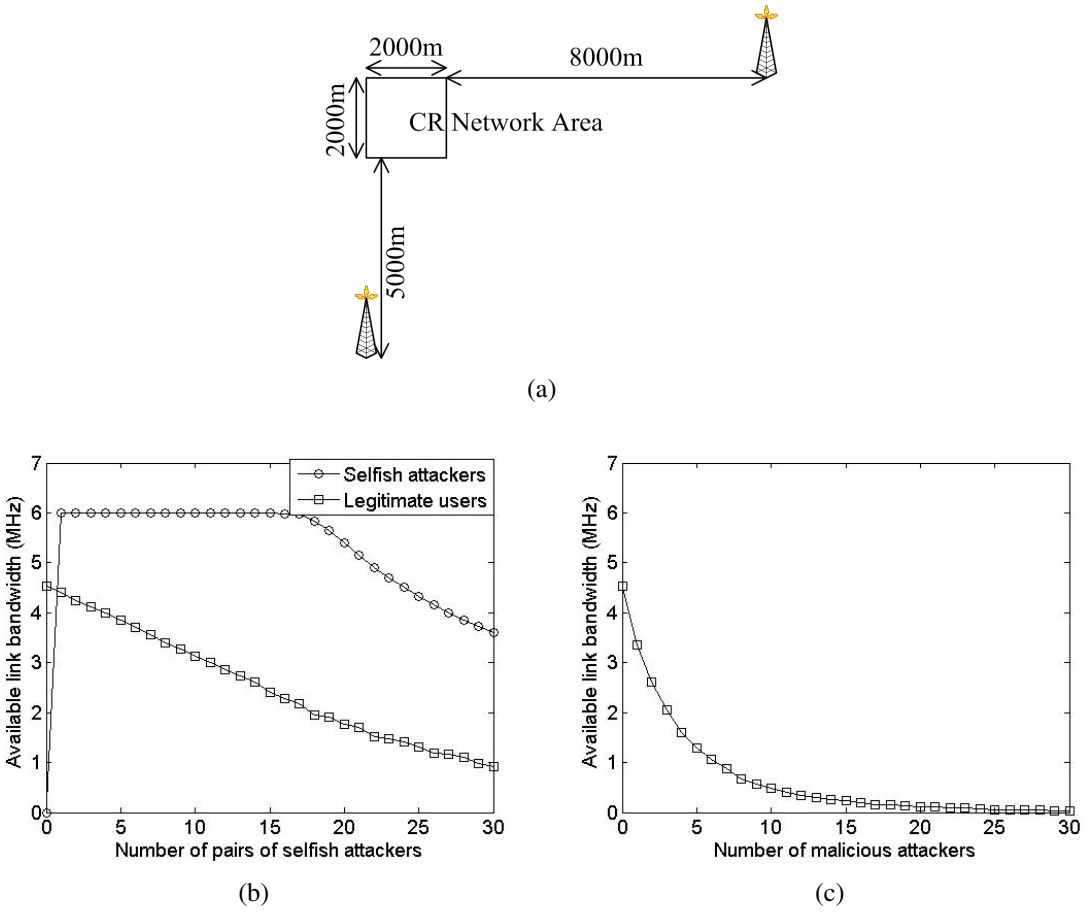


Figure 3.2: Simulation showcasing the effect of PUE attacks. (a) Simulation layout. (b) Effect of selfish PUE attacks. (c) Effect of malicious PUE attacks.

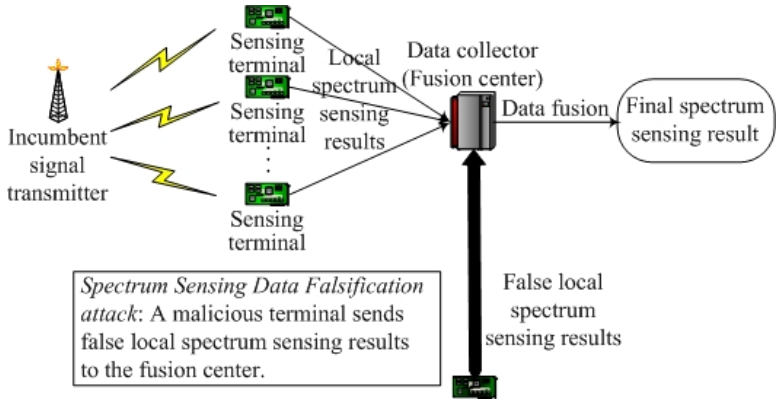


Figure 3.3: A spectrum sensing data falsification attack.

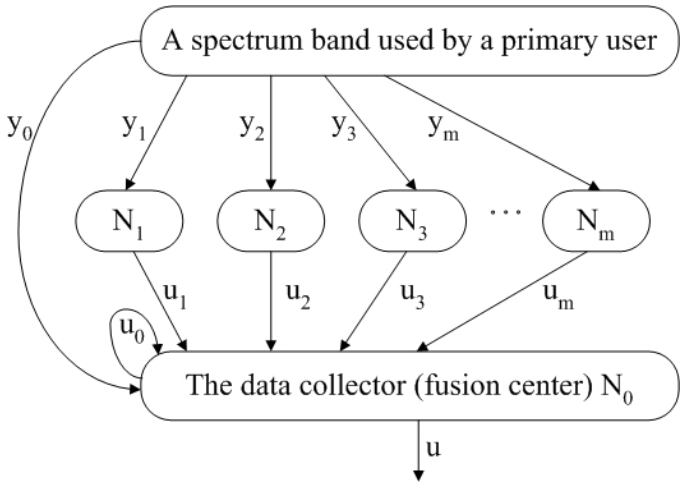


Figure 3.4: Modeling DSS into a parallel fusion network.

sensing terminals) denotes one of N_0's sensing terminals (N_0 is both a data collector and a sensing terminal), y_i represents the primary signal received at N_i, and u_i is the local spectrum sensing report that N_i sends to N_0. The output u is the final sensing decision, which is a binary variable—a "one" denotes the presence of a primary signal, and a "zero" denotes its absence. The data fusion problem therefore can be regarded as a binary hypothesis testing problem with two hypotheses denoted as H_0 and H_1. To simplify the discussions, the following discussion assumes that spectrum sensing is carried out in a single spectrum band and each u_i is also binary.

- *Decision fusion* [52] requires the data collector to sum up all u_i's. A threshold value that is no less than one and no greater than $m+1$ needs to be specified. If the sum of u_i's is greater than or equal to the threshold, then the final sensing decision is "occupied", i.e., $u = 1$ and H_1 is accepted; otherwise the band is determined to be "fallow", i.e., $u = 0$ and H_0 is accepted. Depending on the value of the threshold, decision fusion can have several variants. A threshold value of one is an "OR" fusion rule, a value of $(m+1)$ is an "AND" fusion rule, and a value of $\frac{m+1}{2}$ is a "Majority" fusion rule. Because interference to primary users should be minimized, usually a conservative strategy is favored, which takes a threshold value of

one (i.e., an "OR" fusion rule). In this case, even if a band is free, as long as there is one N_i that erroneously reports $u_i = 1$, the final result will be "occupied," causing a false alarm. If an SSDF attacker exploits this and always reports one as its local spectrum sensing result, then the final result will always be "occupied". To prevent such a scenario, one can increase the threshold value. However, increasing the threshold value has the downside of increasing the probability of miss detection of primary users. Moreover, increasing the threshold is ineffective in decreasing the false alarm probability when there are multiple attackers.

- *Bayesian detection* [40] requires the knowledge of *a priori* probabilities of u_i's when u is zero or one, i.e., $P(u_i|H_0)$ and $P(u_i|H_1)$. It also requires the knowledge of *a priori* probabilities of u, i.e., $P_0 = P[u = 0]$ and $P_1 = P[u = 1]$, respectively. There are four possible cases. In two cases, the sensing decisions are correct, while in the other two cases, the decisions are incorrect. The two incorrect decisions are referred to as miss detection ($u = 0$ when the band is occupied) and false alarm ($u = 1$ when the band is fallow). The two correct decisions (i.e., $u = 0$ when the band is fallow and $u = 1$ when the band is occupied) are associated with small costs and the incorrect ones are associated with large costs. The case of miss detection of a primary user may result in interference to the primary user, and hence this case is least wanted and assigned the largest cost. The overall cost is the sum of the four costs weighted by the probabilities of the corresponding cases. Bayesian detection can be represented by the following test, which outputs a final spectrum sensing decision that minimizes the overall cost:

$$\prod_{i=0}^{m} \frac{P[u_i|H_1]}{P[u_i|H_0]} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P_0(C_{10} - C_{00})}{P_1(C_{01} - C_{11})}, \tag{3.1}$$

where $C_{jk}(j = 0, 1; k = 0, 1)$ is the cost of declaring H_j true when H_k is present. When a

network is under SSDF attacks, the values of the *a priori* conditional probabilities of u_i's are not trustworthy. As a result, Bayesian detection is no longer optimal in terms of minimizing the overall cost.

- *Neyman-Pearson test* [28] does not rely on the knowledge of *a priori* probabilities of u or any cost associated with each decision case. It still requires the knowledge of *a priori* probabilities of u_i's when u is zero or one. Additionally, either a maximum acceptable probability of false alarm or a maximum acceptable probability of miss detection needs to be defined. Neyman-Pearson test guarantees that the other probability is minimized while the defined probability is acceptable. Neyman-Pearson test is represented as

$$\prod_{i=0}^{m} \frac{P[u_i|H_1]}{P[u_i|H_0]} \begin{array}{c} H_1 \\ > \\ < \\ H_0 \end{array} \lambda, \tag{3.2}$$

  where λ is a threshold calculated from the defined probability of false alarm or miss detection. As (3.1) and (3.2) show, Bayesian Detection and Neyman-Pearson test both boil down to a fixed-number likelihood ratio test, and their only difference lies in the way to choose the threshold. As with Bayesian detection, Neyman-Pearson test also requires the knowledge of the *a priori* conditional probabilities of u_i's when u is zero or one. For the same reason discussed above, SSDF attacks would undermine the optimality of the test and potentially cause miss detection or false alarm instances.

The aforementioned data fusion techniques share two properties in common that contribute to their vulnerability to SSDF attacks. First, these techniques treat all sensing terminals indiscriminately, regardless of whether a sensing terminal is reporting true or false sensing data. When an SSDF attacker constantly injects false data, the ideal solution would be to filter the data and only accept

inputs from reliable sensing terminals. Second, both techniques cannot guarantee both a bounded false alarm probability and a bounded miss detection probability.

3.3 Control Channel Jamming Attacks

As discussed in 2.4, jamming attacks that strategically interfere with the transmission or reception of control information in a communication link will significantly increase the attack effectiveness. Particularly for a CR network, because secondary users can typically access multiple opportunistic channels using DSA, it might be difficult for a jammer to blindly choose a channel to attack. Instead, if there is a channel containing all control frames, a wiser jammer would aim at disrupting the control channel in CR networks. We call such an attack a CCJ attack. As with jamming attacks in conventional wireless networks, a CCJ attack in CR networks is easy to launch and effective in wreaking havoc. Unfortunately, the existing cognitive MAC proposals have not fully considered such a threat. In this section, we show why the cognitive MAC schemes discussed in 2.3 are vulnerable to CCJ attacks. We also explain why CR networks may require different solutions to counter CCJ attacks compared to the countermeasures discussed in 2.4.

Before analyzing the details of the effect of a CCJ attack in CR networks, we need to first make some assumptions about the jammer. If in a jamming attack a jammer had infinite power or the number of jammers were infinite, then defending against the attack would be impossible. However, in reality, the capability of an adversary is never "infinite". Moreover, we consider the CR network application to be for non-critical, commercial use, because more critical network applications such as military communications typically require exclusive spectrum access instead of relying on DSA. In such a network environment, the jammer is likely to hide its identify for its own benefit. If an adversary node launches attacks without concealing its identity, the node will be easily identified

and be removed from the network. Therefore, we assume that a jammer in a CCJ attack is identical to any secondary user in a CR network in terms of the functionality (e.g., modulation type and transmission range). The difference lies in the behavior: the jammer will try every means to interfere with the control information exchange of a cognitive MAC protocol in its neighborhood.

The fundamental reason that a CCJ attack would be effective against the distributed cognitive MAC protocols discussed in 2.3 is that these protocols use a control channel to exchange control information. Depending on their implementation, the distributed protocols can be divided into two categories: using a common control channel and using a local control channel. The schemes proposed in [34, 41, 53, 54] all use a common control channel to exchange control information among secondary users. Such a design is known to be vulnerable against jamming attacks—jamming the common control channel would simply disable the entire CR network. Using a local control channel is a more robust solution. In [81], each group of secondary users that are closely located use distributed coordination to elect a local control channel (a.k.a. coordination channel), thus avoid using a common control channel for the whole network. Although this approach, compared to the common control channel design, is more robust against jamming attacks, the improvement is limited. Note that we have assumed that a jammer is indistinguishable from legitimate secondary users and has limited transmission range. Therefore, each jammer is able to jam the control channel only in its neighborhood and disabling a whole CR network requires multiple jammers to be distributed across the network. In a local control channel design, a "common" control channel still exists in a neighborhood. As long as a jammer is placed at the center of the neighborhood, the same disruptive effect as that in the common control channel design can be achieved.

Note that in some research works no control channel is mentioned [12, 55, 77, 80]. This is not because the proposed schemes do not use a control channel, but because their focus is in optimization of maximizing network throughput, promoting link fairness, or minimizing packet collision.

Therefore, the operational MAC issues such as packet scheduling were ignored. To make these schemes full-fledged, it is still necessary to add the MAC functions (which should at least include neighbor discovery, channel scheduling, and packet scheduling) to support control information exchange.

Once a control channel (irrespective of whether it is a common or local control channel) is known to a jammer, the jammer has several options to launch a CCJ attack. Some proposed schemes [34, 53, 54] have inherited and extended the 802.11 MAC. In the 802.11 MAC, a sender and a receiver exchange RTS and CTS frames before using DATA and ACK frames to exchange data. There are at least four known ways to launch a CCJ attack against such a MAC. First, a CCJ attacker may try to corrupt the RTS and CTS of other users to prevent data transmission. Second, the attacker may send many spurious RTS/CTS frames without transmitting data or set large duration values in the NAV (Network Allocation Vector) field of RTS/CTS frames to preempt the control channel. Third, the attacker may corrupt ACKs to cause the ACK contention window to increase, leading to larger backoff or transmission failure. Fourth, the attacker may corrupt the Cyclic Redundancy Check (CRC) field (32 bits for 802.11) in any frame to cause transmission failure. The schemes in [41, 81] used different techniques, but they are still subject to certain types of CCJ attacks. For example, if any frame contains a checksum field, it becomes an objective of a CCJ attack. The scheme in [41] uses a busy tone, which may also be used by a CCJ attacker to jam the control channel. In [81], all secondary users are synchronized and tuned to the control channel during a periodic coordination window (CHWIN). An attacker can maximize the effect of its attack by launching a jamming attack during this window.

As discussed above, most existing cognitive MAC protocols are vulnerable to CCJ attacks. On the other hand, the countermeasures against jamming attacks in existing literature are not easily applicable to CCJ attacks in CR networks. For example, DSSS and FDSS have seen very little

application in today's many distributed, low-cost wireless networks [73]. The idea of channel hopping might be effective, which is relatively easy to implement in multi-channel 802.11 since the number of available channels is fixed. In CR networks, the availability of a channel dynamically changes with primary user activities. Therefore, the synchronization between a sender and a receiver becomes a much more challenging problem.

3.4 Chapter Summary

This chapter discussed three attacks at the physical layer and link layer of CR networks: PUE attacks, SSDF attacks, and CCJ attacks. A PUE attack belongs to a physical-layer attack. An SSDF attack is launched at the link layer (because it requires using the link layer to transmit local spectrum sensing data), but it takes effect at the physical layer. Both PUE attacks and SSDF attacks can disrupt spectrum sensing in a CR network. A CCJ attack is a link-layer attack. Note that while a jamming attack is typically a physical-layer attack, a CCJ attack is considered a link-layer attack because the concept of control channel belongs to the link layer. An CCJ attack disrupts the MAC in a CR network. Depending on the cognitive MAC protocol being used, a CCJ attack can be launched with many options. While PUE attacks and SSDF attacks only exist in CR networks, CCJ attacks can be launched similarly in conventional wireless networks but pose a more difficult problem and require a different solution. These attacks compose great security threats that must be effectively contained in a real deployment of a CR network.

Although these attacks are menacing to the normal operation of CR networks, little research has been done to seek their countermeasures. From Chapter 4 to Chapter 6, we present the potential defense solutions to the aforementioned three attacks, respectively.

Chapter 4

A Transmitter Verification Scheme for Detecting PUE Attacks

This chapter discusses a countermeasure against PUE attacks. While it is difficult to devise a defense scheme that completely solves the problem, we focus on the detection of PUE attacks. For this purpose, this chapter proposes a transmitter verification scheme. It utilizes the location information of primary users together with their signal's energy level to verify the legitimacy of a primary signal transmitter. The core of the transmitter verification scheme is the solution to a non-interactive location verification problem or a non-interactive localization problem. This chapter details solutions to both problems.

This chapter begins with Section 4.1 that introduces the procedure of the proposed transmitter verification scheme. Section 4.2 presents and analyzes two non-interactive location verification schemes that can be integrated into the transmitter verification scheme. These verification schemes are evaluated using simulation in Section 4.3. Section 4.4 proposes a non-interactive localization scheme that can also be used in the transmitter verification scheme. The localization scheme is

simulated in Section 4.5. This chapter ends with a summary in Section 4.6.

4.1 A Transmitter Verification Scheme

Before describing the proposed transmitter verification scheme for spectrum sensing, we state some of the assumptions that form the foundation of the scheme. The primary user is assumed to be a network composed of TV signal transmitters (i.e., TV broadcast towers) and receivers. A TV tower's transmitter output power is typically hundreds of thousands of Watts [75], which corresponds to a transmission range from several miles to tens of miles. We assume that the secondary users, each equipped with a hand-held CR device, form a mobile ad hoc network. Each CR is assumed to have a maximum transmission output power that is within the range from a few hundred milliwatts to a few watts—this typically corresponds to a transmission range of a few hundred meters. An attacker, equipped with a CR, is capable of changing its modulation mode, frequency, and transmission output power.

Based on the above assumptions, we propose a transmitter verification scheme for spectrum sensing that is appropriate for hostile environments; the transmitter verification scheme is illustrated in Fig. 4.1. In the network model under consideration, the primary signal transmitters are TV broadcast towers placed at fixed locations. Hence, if a signal source's estimated location deviates from the known location of the TV towers and the signal characteristics resemble those of primary user signals, then it is likely that the signal source is launching a PUE attack. An attacker, however, can attempt to circumvent this location-based detection approach by transmitting in the vicinity of one of the TV towers. In this case, the signal's energy level in combination with the signal source's location is used to detect PUE attacks. It would be infeasible for an attacker to mimic both the primary user signal's transmission location and energy level since the transmission power

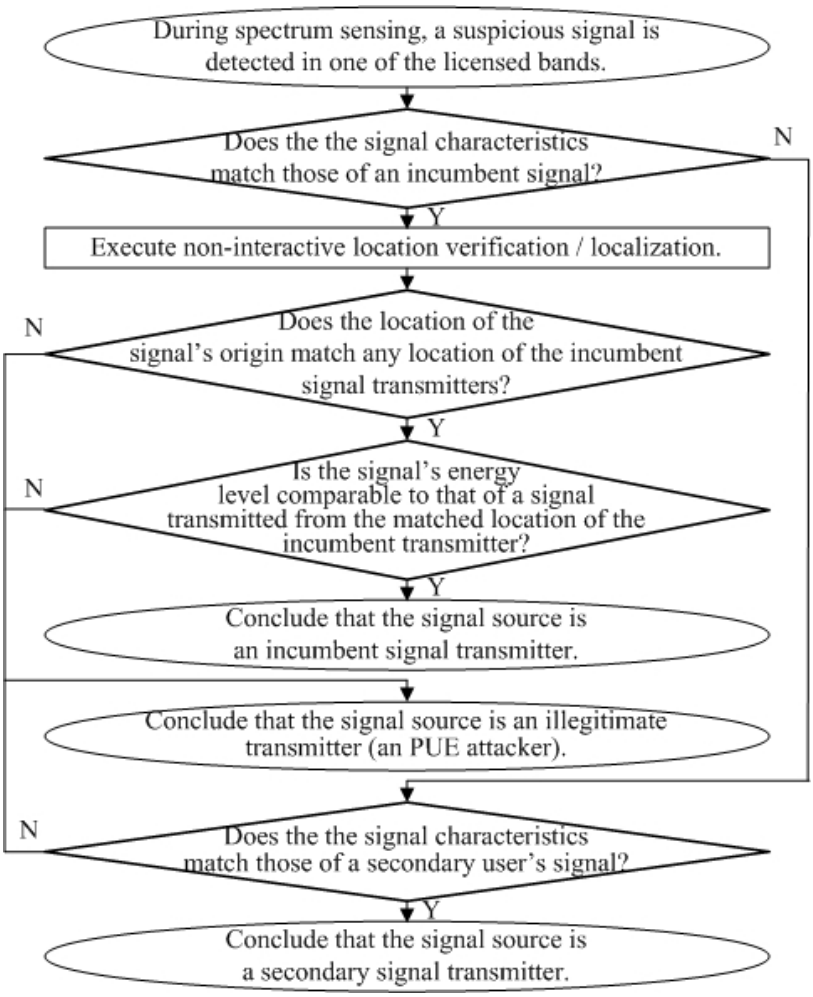


Figure 4.1: A flowchart of the transmitter verification scheme for spectrum sensing.

of the attacker's CR is several orders of magnitude smaller than that of a typical TV tower. Once an instance of a PUE attack has been detected, the estimated signal location can be further used to pinpoint the attacker.

As discussed previously, the key aspect of the transmitter verification scheme is the verification/estimation of the transmitter location. This problem—called by various names such as location estimation, location identification, localization, positioning etc.—has been studied extensively in the past. Section 2.5 has provided a brief summary of existing research on this problem. The primary signal transmitter localization problem, however, is more challenging for two reasons.

First, the following requirement must be met: *no modification should be made to primary users to accommodate the DSA of licensed spectrum*. Because of this requirement, including location information in a primary user's signal is not a viable solution. The requirement also excludes the possibility of using a localization protocol that involves the interaction between a primary user and the localization device(s). Thus, the primary signal transmitter localization problem becomes a *non-interactive* localization problem. Second, it is the transmitter but not the receiver that needs to be localized. When a receiver is localized, one does not need to consider the existence of other receivers. However, the existence of multiple transmitters may add difficulty to transmitter localization. In the following sections, we discuss two type solutions to the non-interactive primary signal transmitter localization problem. The first type solution is a non-interactive location verification scheme and the second type solution is a full-fledged localization scheme. While the former scheme only verifies whether a detected signal source's location (without knowing the actual location) matches its supposed position, the latter scheme could derive the location of a detected signal source.

4.2 Location Verification Schemes

In this section, we devote our discussions to the techniques to realize non-interactive location verification. We focus on two different techniques. The first one is called *Distance Ratio Test* (DRT), which utilizes the RSS of a signal source. The other one is called *Distance Difference Test* (DDT), which relies on the received signal's relative phase difference when the signal is received at different receivers.

The following assumptions need to be made to support the operations of DRT and DDT. We assume that trusted location verifiers (LVs) exist for performing DRT or DDT. An LV can be a dedicated

node, a secondary user with enhanced functions (to carry out DRT/DDT), or a fixed/mobile base station. We assume that the area spanned by the CR network is populated with two types of LVs: one or more master LVs and slave LVs. A master LV has a database of the coordinates of every TV tower whose signal reaches the area spanned by the CR network. Each LV is assumed to know its location from a secure GPS system [35]. In addition, we assume that all of the LVs are synchronized and can communicate with each other through some control channel. In the following discussions, we restrict our discussions to two-dimensional localization.

4.2.1 Distance Ratio Test (DRT)

RSS-based localization has been discussed in 2.5. For radio systems that use tall towers, such as TV systems, the two-ray ground reflection model has been found to be reasonably accurate for predicting large-scale signal strength [57]. The model is represented as follows

$$RSS = P_t G_t G_r \frac{h_t^2 h_r^2}{d^4 L} \tag{4.1}$$

where P_t is the transmitted signal power, G_t and G_r are the antenna gains of the transmitter and the receiver, respectively, h_t is the height of the transmitter, h_r is the height of the receiver, d is the propagation distance, and L is other system loss.

In a hostile environment, parameters such as P_t, G_t, and h_t can be readily manipulated by an attacker launching a PUE attack. Thus, DRT employs a cooperative distance ratio verification scheme, which is independent of those parameters.

In a single iteration of DRT, a pair of LVs, represented by LV_1 and LV_2, simultaneously measure the RSS of a signal in the band of interest, obtaining results R_1 and R_2, respectively. The two LVs are assumed to be identical with respect to the parameters of (4.1) except for their distances to the signal source. Suppose that the positions of LV_1 and LV_2 are (x_1, y_1) and (x_2, y_2), respectively.

The values of R_1, R_2, (x_1, y_1), and (x_2, y_2) are sent to a master LV (note that LV_1 or LV_2 or even another LV may act as a master LV). After receiving the parameters, the master LV goes through the following procedure for each TV tower's coordinate in its database.

1. Suppose that the two dimensional coordinate of the first TV tower is (u_1, v_1). The master LV calculates the reference distance ratio as:

$$\rho = \frac{\sqrt{(x_1 - u_1)^2 + (y_1 - v_1)^2}}{\sqrt{(x_2 - u_1)^2 + (y_2 - v_1)^2}}. \tag{4.2}$$

2. The master LV calculates the *measured* distance ratio, given by the following equation, using the RSS measurements:

$$\rho' = \frac{d_1}{d_2} = \sqrt[4]{\frac{R_2}{R_1}}, \tag{4.3}$$

   where d_1 and d_2 are the respective distances between LV_1 and the signal source and LV_2 and the signal source.

3. The master LV checks whether

$$\rho' \in [\frac{\rho}{(1 + \varepsilon_1)}, (1 + \varepsilon_1)\rho], \tag{4.4}$$

   where $\varepsilon_1 (\geq 0)$ is the expected maximum error; it includes both measurement error and modeling error.

If (4.4) does not hold, the signal source under scrutiny fails the location verification for the TV tower used in Step 1; otherwise, it passes the location verification. The above steps are repeated using the coordinates of the next TV tower, and the process is repeated until all of the coordinates in the database have been exhausted. If the signal source fails all of the location verifications, then the master LV concludes that the location of the signal source is not consistent with any of the TV towers in its database.

The practicality of DRT hinges on its accuracy. If an attacker is at a location that induces a similar distance ratio as that of a primary signal transmitter, the DRT may fail to recognize the signal as an attacker's signal, resulting in a false negative instance. On the other hand, if ε_1 is too small, DRT may mistakenly identify a primary signal as an attacker's signal, resulting in a false positive instance. To increase DRT's accuracy, multiple DRT iterations must be performed, each iteration using a different pair of LVs.

There are two caveats about the DRT that should be noted. First, since DRT relies on a large-scale propagation model, the possible fluctuations in RSS caused by small-scale fading are not considered. The effects of small-scale fading may vary the RSS by as much as three or four orders of magnitude when a receiver's position changes by only a fraction of a wavelength [57]. To effectively mitigate such effects, an "averaged" RSS value should be used—i.e., RSS should be averaged over multiple measurements made within a surrounding range of 5λ to 40λ [57], where λ is the wavelength of the signal. For TV signals transmitted at UHF 617MHz, this means that an LV needs to average multiple synchronous RSS measurements over a range of 2.5m to 20m. This approach, however, could be expensive to implement in practice. Second, DRT does not consider the fact that the radio propagation model is affected by various environmental variables. Different propagation environments may require the use of different parameters, and may even require the use of totally different propagation models. Recall that in DRT, the two LVs use the identical radio propagation model. This approach can result in erroneous location verification results if the two radio propagation paths from the signal source to each LV go through significantly different environments. Addressing such cases would require significant changes to the DRT technique.

4.2.2 Distance Difference Test (DDT)

We propose an alternate technique to DRT, namely DDT, that verifies the difference in the two distances between a primary user and a pair of LVs. The difference in distance can be measured by measuring the phase shift of a signal at the two LVs. DDT does not suffer from DRT's drawbacks.

Analog TV signals have embedded synchronization pulses. In particular, such a pulse periodically appears every 64μs , with a maximum deviation of 0.25μs [69]. For digital TV systems, each symbol spans 224μs, in which 7μs is a silent period for inter-symbol separation [69]. If the primary signals are analog TV signals, the distance difference between a signal source and two LVs can be estimated by calculating the time difference in which each LV sees the same synchronization pulse. The time difference is readily converted to distance difference by multiplying the speed of light to the time difference. If the primary signals are digital TV signals, the time difference in which each LV sees the rising (or falling) edge of the same symbol is used.

Fig. 4.2 shows how the time difference is measured when primary signals are analog TV signals. In the figure, two synchronized LVs, LV_1 and LV_2, simultaneously record the time at which they see the synchronization pulse of the TV signal, and record the time values as t_1 and t_2, respectively. The time difference is calculated as $t_\Delta = t_1 - t_2$. Suppose that the coordinates of LV_1 and LV_2 are (x_1, y_1) and (x_2, y_2), respectively. The values of t_1, t_2, (x_1, y_1), and (x_2, y_2) are sent to the master LV. After receiving the parameters, the master LV goes through the following procedure for each TV tower's coordinate in its database.

1. Suppose that the two dimensional coordinate of the first TV tower is (u_1, v_1). The master LV calculates the reference distance difference as:

$$s = \sqrt{(x_1 - u_1)^2 + (y_1 - v_1)^2} - \sqrt{(x_2 - u_1)^2 + (y_2 - v_1)^2}. \tag{4.5}$$

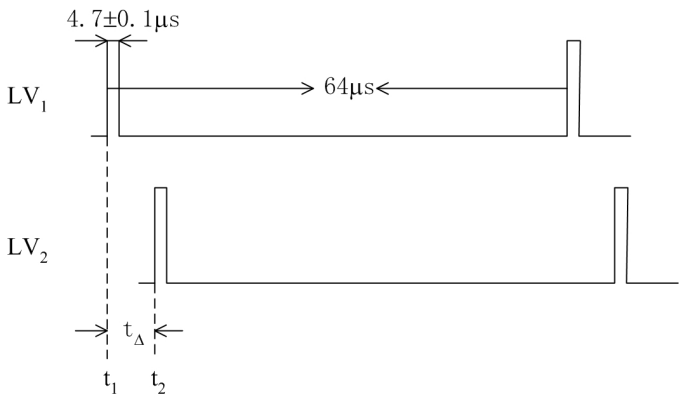


Figure 4.2: The measurement of time gap for DDT.

2. Then the master LV calculates the observed distance difference using the time difference:

$$s' = c(t_1 - t_2) = ct_\Delta, \tag{4.6}$$

where c is the speed of light.

3. The master LV checks whether

$$s' \in [s - c\varepsilon_2, s + c\varepsilon_2], \tag{4.7}$$

where $\varepsilon_2 (\geq 0)$ is the expected maximum time measurement error.

If (4.7) does not hold, the signal source under scrutiny fails the location verification for the TV tower used in Step 1; otherwise, it passes the location verification. The above steps are repeated using the coordinates of the next TV tower, and the process is repeated until all of the coordinates in the database have been exhausted. If the signal source fails all of the location verifications, then the master LV concludes that the location of the signal source is not consistent with any of the TV towers in its database.

In the above discussions, we have neglected to discuss a very important aspect of DDT's feasibility. If the temporal separation between two consecutive synchronization pulses (or symbols in case of

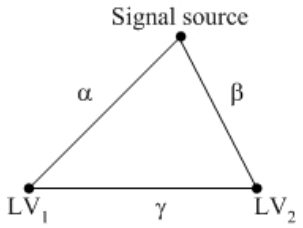


Figure 4.3: DDT is feasible if $\gamma < \delta \cdot c/2$.

digital TV signals) is too small, the DDT scheme may be infeasible. Suppose that the separation between pulses, represented by δ, is small enough for the relation ($t_\Delta \geq \delta/2$) to hold. In this case, it is nearly impossible for two LVs to make sure that they are recording the time of the same pulse since the time instants in which the two LVs see the same pulse may be separated by more than the length of the time duration in which each of them observes a different pulse. The value t_Δ is determined by the difference between the lengths of the two (line-of-sight) paths: one path from the signal source to LV_1, which we represent as α, and the other path from the signal source to LV_2, which we represent as β. In order for DDT to be feasible, this distance difference must be small enough so that the relation $|\alpha - \beta| < \delta \cdot c/2$ is satisfied. See Fig. 4.3 for an illustration. Due to the triangle inequality theorem, the distance difference is always less than the distance between the two LVs, which we represent as γ. Hence, as long as the distance between the two LVs is small enough to satisfy $\gamma < \delta \cdot c/2$, DDT is feasible. For example, in an analog TV system, two consecutive synchronization pulses are separated by 64μs, which is equivalent to 19,200m spatial separation. As long as the two LVs are less than 9,600m away from each other, DDT is feasible.

4.2.3 Security Analysis

For DRT and DDT to be effective in hostile environments, several security issues need to be addressed. In this subsection, we focus on two key problems that impact the security and reliability of DRT and DDT. The first problem is ensuring the robustness of the location verification process

against attacks, and the second problem is ensuring secure data exchange among LVs.

4.2.3.1 Location Verification Scheme's Robustness against Attacks

There is a possibility that a PUE attacker may strategically position its transmitters and adjust their transmission power to circumvent the location verification procedure carried out by the LVs. Such an attack is possible only when an attacker has knowledge about the LVs' location. With the LVs' location information, an attacker can estimate the RSS and the time-of-flight of the signals emitted by its transmitters when those signals reach the LVs. Recall that DRT and DDT utilize RSS and time-of-flight respectively to gauge the location of the signal source. Armed with such estimates, it is possible for an attacker to launch a PUE attack without failing the location verification tests carried out by the LVs.

A straightforward and effective countermeasure to such attacks is to use covert LVs. Here, covert LVs are LVs whose positions are known only to the authority controlling the location verification process. Note that the use of covert verifiers (or base stations) in secure localization schemes is not new (e.g., see [9]). To maintain the LV's covertness while not affecting communications among the nodes in a CR network, existing protocols for anonymous communications (such as MASK [78]) can be used.

An attacker may try to disrupt the location verification procedure by synchronizing its transmitters to send their signals simultaneously. In such a case, the LVs would receive a mixture of multiple signals. This, however, does not help the attacker's transmitters pass the transmitter verification process (see Fig. 4.1). For instance, the aggregate of the signals sent by a group of malicious transmitters will have synchronization pulses at irregular intervals (due to the overlapping of multiple signals). Such a signal deviates from the characteristic of a legitimate analog TV signal, and therefore would be readily identified as a non-primary signal.

4.2.3.2 Secure Data Exchange among LVs

Another security concern is the security of the data exchange between slave LVs and the master LV. The exchanged data must be encrypted and authenticated to avoid eavesdropping, insertion, modification, or replay attacks carried out by attackers.

The following protocol utilizes public-key cryptosystems to secure the messages exchanged between the master LV and slave LVs. We assume the existence of a PKI (public-key infrastructure) that takes care of key distribution, renewal, and revocation. The master LV initiates the protocol by broadcasting the following message to the slave LVs:

$\{ID, E_{master-LV}[t_s, FLG, ID-1, D_{LV-1}[ID-1, B, t], ID-2, D_{LV-2}[ID-2, B, t],$

$\cdots, ID-K, D_{LV-K}[ID-K, B, t]]\}$,

where ID indicates the master LV's identity, $E_{master-LV}[]$ denotes an encryption operation using the master LV's private key, t_s is a timestamp, FLG is a flag indicating whether DRT or DDT should be carried out, $ID-i$ ($i = 1, \cdots, K$, where K is the number of slave LVs) denotes the identity of a slave LV, $D_{LV-i}[]$ represents an encryption operation with a slave LV's public key, B denotes the spectrum band in which location verification should be carried out, and t is the start time for the location verification procedure. Note that although the broadcast message reveals the sender's identity, it does not necessarily reveal the sender' position when an anonymous communications protocol is employed [78]. A slave LV that has received the broadcast message decrypts the appropriate portions of the message with its own private key and the master LV's public key. According to the information revealed in the decrypted message, the LV either measures the RSS (when DRT is indicated) or records the time of the first two consecutive pulses or rising/falling edges (when DDT is indicated) observed after time t. Suppose that the master LV has instructed the slave LVs to measure the RSS of a particular signal. A slave LV replies to the master LV with the following message:

$\{ID - i, E_{LV-i}[t_s], D_{master-LV}[E_{LV-i}[B, t, t_a, x_{ID-i}, y_{ID-i}, P_{ID-i}]]\}$,

where $ID - i$ is the sending slave LV's identity, $E_{LV-i}[]$ denotes an encryption operation using the sender's private key, $D_{master-LV}[]$ denotes an encryption operation with the master LV's public key, (x_{ID-i}, y_{ID-i}) is the position of the sender, and P_{ID-i} is the RSS measurement value in band B at time t_a, which is the time when the signal is first observed. If the master LV had instructed the LVs to record the times of the first two consecutive pulses (or two consecutive rising edges/falling edges of symbols when the signal source is transmitting digital TV signals), then the LV replies with the message

$\{ID - i, E_{LV-i}[t_s], D_{master-LV}[E_{LV-i}[B, t, x_{ID-i}, y_{ID-i}, t_{ID-i-1}, t_{ID-i-2}]]\}$.

The above message replaces P_{ID-i} and t_a with t_{ID-i-1} and t_{ID-i-2}, which are the times when the first two consecutive synchronization pulses are seen. Two measurements are required because DDT requires that two LVs measure the same pulse. If only one measurement is taken starting from time t, then two LVs may be measuring two different pulses. When two consecutive measurements are taken, as long as two LVs are distanced closer than what the signal can travel within the time period between two consecutive pulses, there is at least one pulse that the two LVs have both measured (i.e., the pulse that is received at the two LVs within the time interval of $\delta/2$, as explained in 4.2.2). After receiving the messages from the slave LVs, the master LV carries out either DRT or the DDT as described in 4.2.1 and 4.2.2.

4.3 Simulation of the Location Verification Schemes

In this section, we present the simulation results for DRT and DDT. In particular, we focus on the impact of measurement error on the false negative ratio, which represents the probability of a PUE attacker passing location verification.

4.3.1 Simulation Settings

The network layout used in the simulations is shown in Fig. 4.4. The CR network is located within a 2000m$\times$2000m square area A_1. The primary signal transmitter, a TV tower, is located at either position L_1 or L_2 in the figure. The location L_1 represents the scenario in which the transmitter is within the area spanned by the CR network, and the location L_2 represents the scenario in which the transmitter is outside this area. The distance between L_2 and the leftmost border of A_1 is 8000m, which is the maximum transmission range of the TV tower. This value is calculated based on realistic assumptions. Suppose the following parameters: the EIRP of the TV towers (transmitters) is 2500KW, transmitters' effective antenna height is 100m, receivers' effective antenna height is 1m, and receivers' energy detection sensitivity is –94dbm. Under these conditions, one can derive a transmission radius of 8000m using the rural environment version of the HATA model [57]. The position of L_2 is considered for out-of-CR-network primary signal transmitter because in this scenario the expected energy level of the TV tower signal received at the CR network will become very small, thus making it difficult to distinguish the TV tower signal from a secondary user's signal just based on the signal energy level. We assume that a single PUE attacker equipped with a hand-held CR can be located either inside area A_1 or inside area A_2. They represent two possible cases in which the PUE attacker lies either in the CR network or out of the CR network. Note that area A_2 is relatively close to area A_1 because the transmission range of the attacker's CR is rather limited. The placement of A_2 and L_2 on the same side of A_1 represents the worst case—compared to other possibilities, in this case the relative position between the attacker and the LVs will be most similar to the relative position between the TV tower and LVs, so that the distance ratio or distance gap induced by an attacker in A_2 will be more likely to be close to that induced by L_2. Based on the above considerations, we carried out the simulation experiments in four different settings, in which the TV tower could be either inside or outside the CR network, and the PUE attacker could

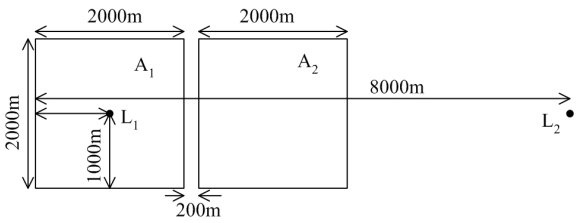


Figure 4.4: The network layout used in the simulations for DRT and DDT.

also be either inside or outside the CR network. Specifically, these settings are:

- Setting 1: the attacker is in A_1 and the primary user is at L_1.
- Setting 2: the attacker is in A_1 and the primary user is at L_2.
- Setting 3: the attacker is in A_2 and the primary user is at L_1.
- Setting 4: the attacker is in A_2 and the primary user is at L_2.

In the simulation of each setting, the attacker's transmitter was placed randomly within the area specified by the setting, and the LVs were placed randomly within in CR network area (i.e., A_1). Each simulation run simulates a transient process in which the PUE attacker transmits a signal and the LVs judges whether the test indicated by (4.4) or (4.7) passes or fails. In every simulation run, we generated 300 random locations for the attacker's transmitter and 100 random locations for each LV. Therefore, it amounts to 30,000 iterations for generating every data point in Figs. 4.5 and 4.6, which shows the average result of the iterations. We observed that the chosen number of iterations guaranteed stable simulation results.

In the simulation, we consider DRT's measurement and modeling error ε_1's value in the range of [0, 1]. As (4.4) shows, this is equivalent to allowing an RSS ratio to vary in a range of 50% to 200% of what a legitimate distance ratio is supposed to be. The discussion in 4.2.1 has shown that RSS measurement error can be controlled by taking multiple measurements within a small surrounding

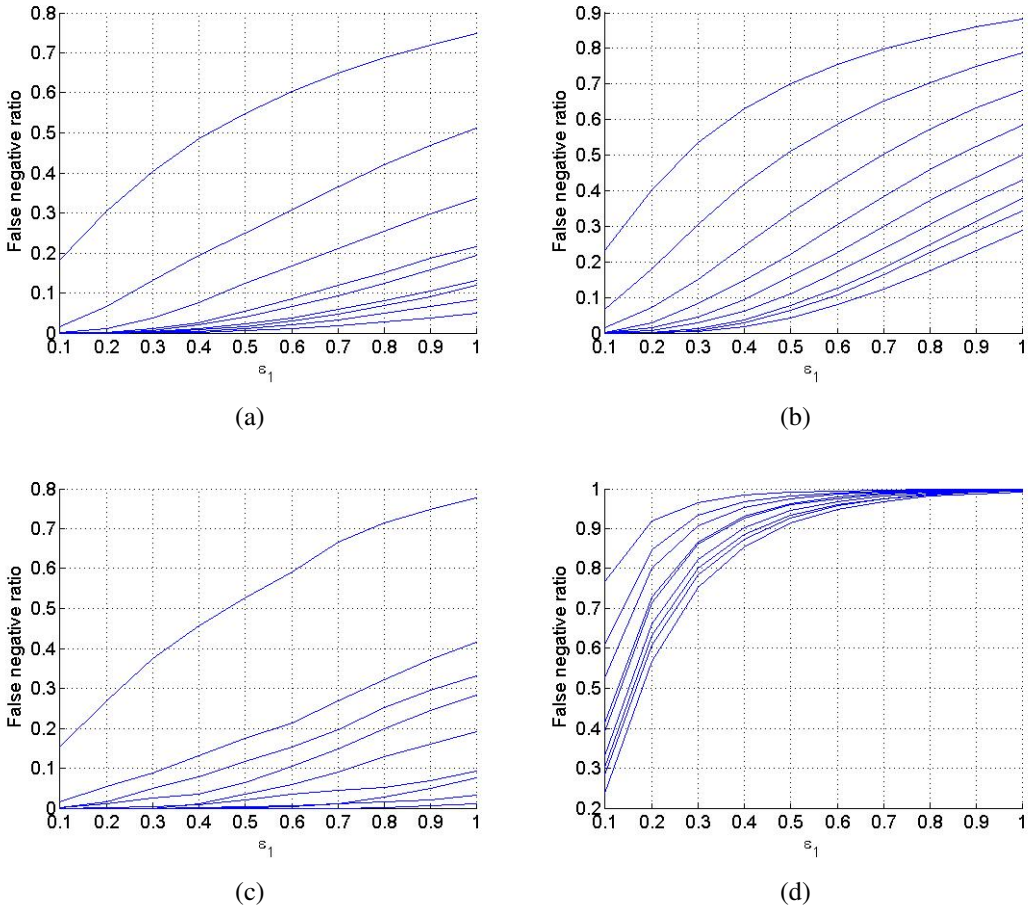


Figure 4.5: DRT simulation results. There are nine curves in each plot. The nine curves, from top to bottom, were obtained by incrementing the number of LVs by one, starting from 2 to 10. (a) Setting 1; (b) Setting 2; (c) Setting 3; (d) Setting 4. The value ε_1 denotes the measurement and modeling error.

range. Therefore, when the path loss model is appropriately chosen, the above RSS ratio range will be large enough to ensure a legitimate primary signal transmitter passing the test. We choose DDT's time measurement error ε_2 in the range of [0, 1μs]. It is known that the current technology supports a maximal time measurement error on the order of a few hundred nanoseconds [69]. Therefore, a maximum value of ε_2 being 1μs is a realistic choice.

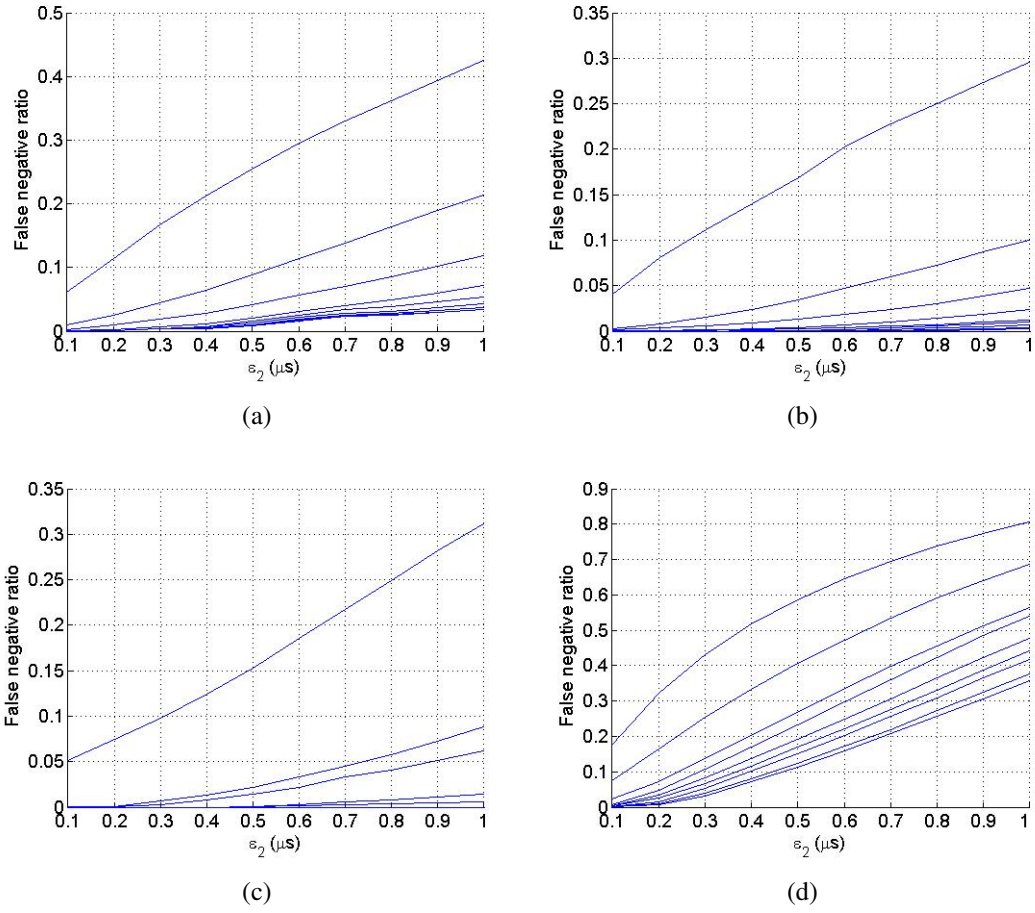


Figure 4.6: DDT simulation results. There are nine curves in each plot. The nine curves, from top to bottom, were obtained by incrementing the number of LVs by one, starting from 2 to 10. (a) Setting 1; (b) Setting 2; (c) Setting 3; (d) Setting 4. The value ε_2 denotes the time measurement error.

4.3.2 Simulation Results

Figs. 4.5 and 4.6 show the simulation results for DRT and DDT, respectively. The false negative ratio is plotted as a function of the error value. As expected, the increase in the number of LVs caused a decrease in the false negative ratio.

The results indicate that the location of the attacker's transmitter relative to the primary signal transmitter has a noticeable impact on the false negative ratio. From Fig. 4.5, we can see that DRT performed poorly in Setting 2 and Setting 4 compared to its performance in the other two settings. The common feature shared by Settings 2 and 4 is that the primary signal transmitter is far away from area A_1 which is where the LVs are located. Hence, irrelevant of which two LVs were chosen, the distance between an LV and the primary signal transmitter would be similar to the distance between the other LV and the primary signal transmitter, thus resulting in a reference distance ratio close to one. In other words, increasing the number of LVs would not contribute significantly to the heterogeneity of the reference distance ratio values. For this reason, increasing the number of LVs, in Settings 2 and 4, did not decrease the false negative ratio dramatically as it did in Settings 1 and 3.

We also notice that DRT showed the poorest performance in Setting 4. This can be attributed to the fact that, in Setting 4, the attacker's transmitter is located in a region that is disjoint with the region that contains the LVs. This would decrease the heterogeneity of the measured distance ratios, thus increasing the false negative ratio even further compared to DRT's performance in Setting 2.

From Fig. 4.6, we can see that DDT's performance is less sensitive to the locations of the attacker's transmitter and the primary signal transmitter.

It should be noted that the false negative ratio values plotted in Figs. 4.5 and 4.6 are only confined to location verification. The other verification procedures in the transmitter verification procedure

(see Fig. 4.1) also need to be considered to derive the overall false negative ratio.

4.4 A Non-interactive Localization Scheme

The location verification schemes proposed in 4.2 can detect PUE attacks under certain conditions. However, they also have two problems. First, the DRT technique heavily depends on a correct radio propagation model and the DDT technique requires expensive devices to measure synchronization pulses with very small error. Second, the detection of PUE attacks using location verification schemes does not reveal any additional information. It is more desirable if the PUE attackers can be localized so that their physical capture could be a potential choice for attack response. In this section, we discuss a localization scheme to provide such a choice. As discussed previously, when localizing a primary signal transmitter, one has to use a non-interactive localization scheme.

Before introducing the proposed localization system, we first discuss how conventional localization techniques (see 2.5) should be improved to address the primary signal transmitter localization problem (which is referred to as the *PSTL problem* hereafter).

Among localization techniques discussed in 2.5, TOA is a receiver-localization technique and needs to be enhanced to support transmitter localization so that it can be applied to the PSTL problem. Such an enhancement is not trivial, especially when one considers the possibility that a malicious transmitter may craft its transmitted signal. TDOA and AOA techniques can both be used for transmitter localization and have relatively high localization precision. To apply them to the PSTL problem, special care must be taken to consider the situations where multiple transmitters or an attacker equipped with a directional antenna exists. The common drawback of both techniques is the requirement of expensive hardware, preventing them from a large-scale deployment. In contrast, RSS-based techniques are more practical for most consumer premise devices in

a CR network. However, for the PSTL problem, one should also consider the issues of possible manipulation of a malicious transmitter or multiple transmitters and the innate inaccuracy of RSS measurement. In the rest of this section, we show that these issues can be addressed by taking many RSS measurements and properly processing the measured RSS data.

4.4.1 Architecture of the Localization System

The basic idea of the proposed localization system uses the fact that the magnitude of an RSS value typically decreases as the distance between the signal transmitter and the receiver increases [27]. Therefore, if one is able to collect a sufficient number of RSS measurements from a group of receivers spread throughout a large network, the location with the peak RSS value is likely to be the location of a transmitter. The advantage of this technique is twofold, when it is used for the PSTL problem: it both obviates modification of primary users and supports localizing multiple transmitters that transmit signals simultaneously.

The requirement to collect RSS distribution in a network naturally leads us to resort to an underlying wireless sensor network (WSN) that can help collect RSS measurements across the network. It should be noted that the idea of using an underlying WSN to facilitate the operation of a CR network is not new. For example, in [64], it was proposed that a spectrum-aware sensor network be used for distributed spectrum sensing, so that the sensor network can provide secondary users with information about spectrum opportunities throughout a network. If sensor nodes in a WSN have the capability to measure RSS and are aware of their positions [27], they can be used to solve the PSTL problem. However, there are two problems that need to be addressed in order for the aforementioned approach to be viable.

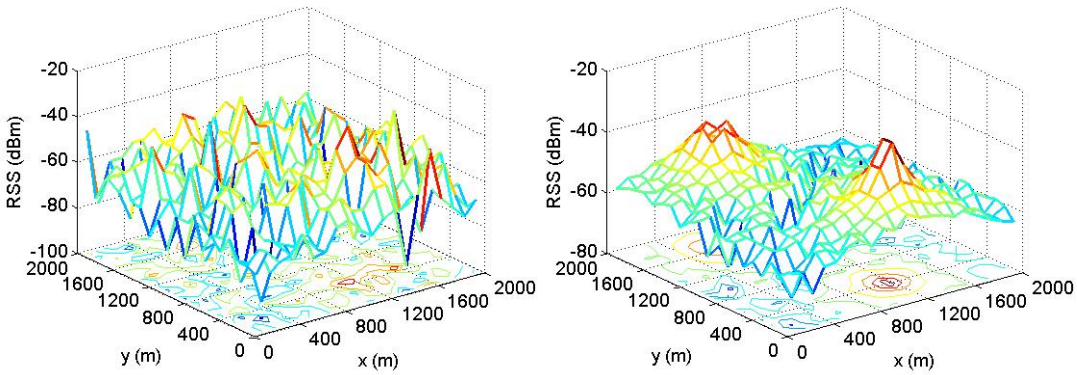
First, path fading may change over time and a PUE attacker may constantly change its location or vary its transmission power to evade localization, thus causing RSS measurements to fluctuate

drastically within a short period of time. This problem cannot be mitigated by taking the average of measurements taken at different times, since the RSS values measured at a given position at different times have different distributions. A possible solution to this problem is to take a "snapshot" of the RSS distribution in a given network, i.e., requiring the sensors of a WSN to take a synchronized RSS measurements in a given band.

The second problem arises from the fact that RSS usually varies by a large magnitude (30dB to 40dB) [57] over short distances. This makes it very challenging to decide the location of primary users just by reading the raw data in a snapshot of RSS distribution. We conducted a simulation experiment to illustrate this problem. A 2000m$\times$2000m network with two transmitters located at (800m, 1800m) and (1300m, 550m) was simulated. Each transmitter's transmission power was 500mW, working at the UHF frequency of 617MHz. The phase shift between the two transmitters was randomly chosen. A statistical log-loss signal propagation model, which was shown to be appropriate for modeling signal propagation behavior in many situations [61], was employed in the simulation. In this model, the expected RSS in decibels is given by:

$$\mu = p + \beta_0 + \beta_1 \ln s, \tag{4.8}$$

where s is the transmitter-receiver distance, p is the transmitted power in decibels, and β_0 and β_1 are constant parameters that need to be calibrated for a specific environment. Note that this is offsite calibration, and no onsite calibration is required [61]. In the offsite calibration, one needs to tune the parameters related to the channel environment (e.g., rural, urban, etc.). Using the model, the distribution of RSS is characterized as a Gaussian random variable with a mean of μ and a variance of σ^2. In [61], a set of parameters approximating real-world results were used, where (β_0, β_1, σ) = (-30.00, -10.00, 10.0). We used the same set of parameters for our simulation. Fig. 4.7(a) shows a snapshot of the RSS in dBm. It can be seen that because of the large variance of the RSS, the snapshot does not reveal obvious RSS peaks (which can be used as approximations for the



(a) (b)

Figure 4.7: RSS distributions obtained from the underlying WSN. (a) A snapshot of the RSS raw-data distribution. (b) The RSS distribution in the network when $\sigma = 0$.

transmitter locations).

However, if the variance can be reduced to a sufficient level, the snapshot would clearly indicate the RSS peaks as illustrated in Fig. 4.7(b). It is therefore reasonable to conjecture that if one is able to decrease the variance using an appropriate *data smoothing* technique, it may be possible to solve the PSTL problem by using the aforementioned localization approach. In the next subsection, we focus on the design of such a data smoothing technique.

4.4.2 The RSS Smoothing Procedure

Data smoothing techniques [65] aim to capture important patterns in raw data, while leaving out noise. By smoothing a snapshot of an RSS distribution in a network, one can decrease the variance in the raw RSS measurements, thus making it possible to identify the RSS peaks.

There are three data smoothing techniques that are usually used to eliminate noise: local averaging, Fourier filters, and loess fitting. In our RSS smoothing problem, robustness against outliers is an important requirement for two reasons. First, the large variance in RSS measurements may result in

a large number of outliers. Second, when an adversarial environment is considered, compromise of sensor nodes may lead to false data injection. Among the three data smoothing techniques, Fourier filters is known to be vulnerable to large variation. Loess fitting requires a large, densely sampled dataset and its robustness against outliers depends on careful design of the weight mechanism used for computing least squares [65]. In contrast, local averaging, especially when the median value is taken, provides the best robustness against outliers. Therefore, we use local averaging, using median values, to smooth RSS measurement data. The details of the smoothing technique are described below.

Without loss of generality, we assume that the coverage area of the WSN is identical to that of a CR network under consideration, which covers an area of $D_x \times D_y$ (m^2). Suppose that we sample a group of "pivot" points that are placed at the intersections of the vertical and horizontal lines of a two-dimensional grid, where each element on the grid is a square with a side of length d. For each pivot point we calculate a "smoothed" RSS value by calculating the median value from the set of RSS measurements collected by neighboring sensor nodes that are located inside an area enclosed by a circle of radius r centered at the pivot point. See Fig. 4.8 for an illustration of how the pivot points are positioned. (Note that the centers of the circles marked with "1's" denote pivot point positions.) Once data smoothing is applied to RSS measurements, one can estimate the positions of the primary signal transmitters by identifying the positions of the pivot points that generate "peak" median values.

Next we discuss the details of how to set the values of r and d and how to identify RSS peak values. We discuss these problems in the context of the statistical log-loss propagation model described in Subsection 4.4.1. We assume that the values of β_1 and σ have been estimated[1], and the density of the sensor nodes is ρ (m^{-2}). Suppose that a primary signal transmitter S is transmitting inside an area defined by a circle of radius r centered at pivot point x. Our objective is to derive the values

[1] As mentioned before, the two values can be estimated using the offsite calibration technique presented in [61].

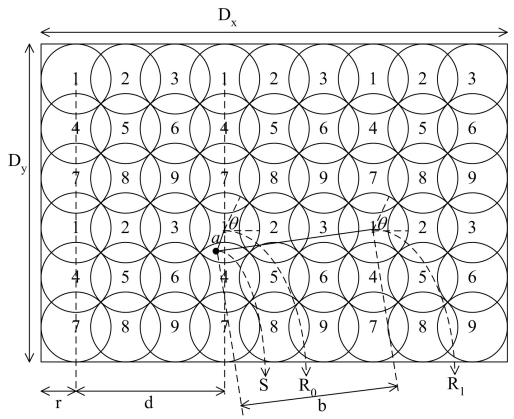


Figure 4.8: Using local averaging to smooth RSS measurement.

of r and d so that the median RSS value calculated from x is greater than the RSS value calculated from its neighboring pivot points, which are located at a distance of d from point x, by at least mdB at a confidence level of P.

Suppose the circular region of radius r centered at x is R_0, and let R_1 denote a circular region centered at a neighboring pivot point that is at a distance of d (see Fig. 4.8). In R_0, the expected RSS after averaging[2] is:

$$\mu_0 = E[p + \beta_0 + \beta_1 \ln s] = p + \beta_0 + \beta_1 \cdot \frac{1}{\pi r^2} \int_0^{2\pi} \int_0^{r} \ln \sqrt{(r\cos\theta + a)^2 + (r\sin\theta)^2} r dr d\theta, \quad (4.9)$$

where a is the distance between the transmitter to the center of R_0. Because $a \leq r$ and $\beta_1 < 0$, it holds that

$$\begin{aligned} \mu_0 &\geq p + \beta_0 + \beta_1 \cdot \tfrac{1}{\pi r^2} \int_0^{2\pi} \int_0^{r} \ln \sqrt{(r\cos\theta + r)^2 + (r\sin\theta)^2} r dr d\theta \\ &> p + \beta_0 + \beta_1 \ln \tfrac{1}{\pi r^2} \int_0^{2\pi} \int_0^{r} \sqrt{(r\cos\theta + r)^2 + (r\sin\theta)^2} r dr d\theta \\ &= p + \beta_0 + \beta_1 \ln \tfrac{8r}{3\pi}. \end{aligned} \quad (4.10)$$

Similarly, in R_1, the expected RSS after averaging is

$$\mu_1 = p + \beta_0 + \beta_1 \cdot \frac{1}{\pi r^2} \int_0^{2\pi} \int_0^{r} \ln \sqrt{(r\cos\theta + b)^2 + (r\sin\theta)^2} r dr d\theta, \quad (4.11)$$

[2]Note that for a random variable with Gaussian distribution, its median is equal to its mean.

where b is the distance between the transmitter and the center of R_1. It holds that $b \geq d - r$. If we further assume that $d > 2r$, it is obvious that

$$\mu_1 < p + \beta_0 + \beta_1 \ln(d - 2r). \tag{4.12}$$

The results from (4.10) and (4.12) enable us to calculate a lower bound of the difference between the medians of measured RSS values in R_0 and R_1:

$$\Delta\mu = \mu_0 - \mu_1 > |\beta_1| \ln[\pi(\frac{3d}{8r} - \frac{3}{4})]. \tag{4.13}$$

Because the RSS measurement can be modeled as a Gaussian random variable [39, 61], $\Delta\mu$ is also a Gaussian random variable. It has a mean of $\mu_0 - \mu_1$ and a variance of $2\sigma^2/(\rho\pi r^2)$. To obtain a sufficient condition that guarantees the difference is larger than m at a confidence level P, we first define a variable x_0 that satisfies:

$$Q(x_0) = 1 - P, \tag{4.14}$$

where the Q-function represents the right-tail probability of a normalized Gaussian variable. Then based on the properties of Gaussian variables, a sufficient condition can be derived:

$$\frac{|\beta_1| \ln[\pi(\frac{3d}{8r} - \frac{3}{4})] - m}{\sqrt{2}\sigma} \cdot \sqrt{\rho\pi} r \geq x_0. \tag{4.15}$$

The values of d and r should satisfy the above condition so that solving the PSTL problem becomes equivalent to finding the circular regions whose median RSS value is at least mdB greater than those in its neighboring regions that are at a distance of d.

Because the above derivation assumed $d > 2r$, the sampled regions cannot cover the entire coverage area of the WSN. Therefore, multiple rounds of sampling should be performed. Fig. 4.8 shows how multiple rounds (nine in total) of sampling cover the whole area. It is obvious that the total number of rounds will be $O = \left\lceil d/(\sqrt{2}r) \right\rceil^2$ and the distance between two neighboring samplings will be d/O.

After all rounds of samplings are finished, a set of regions are identified to have a median RSS value that is at least mdB greater than those in its neighboring regions sampled in the same round. This set, R, indicates the approximate locations of the primary signal transmitters. Next we need to sample more points within the sets to obtain more precise locations. The following steps are executed for this purpose:

1. Group the regions in R into a minimum number of mutually exclusive sets R_1, R_2, ..., R_T so that all regions in each set R_v ($v = 1, 2, \ldots, T$) are interconnected.

2. For the area covered by each R_v, sample all points that are horizontally or vertically apart by $(w \cdot \Delta)$, where w is an integer and Δ is a sample interval determined by the sensor density ρ (see below). For each sampled point, the median is calculated for the RSS measurements over a circular region centering the point with radius r. The location of the point with the maximum median RSS value in R_v is the estimated location of a primary signal transmitter.

Now, we explain how to decide the value of Δ. The value of Δ needs to be sufficiently large so that computation overhead is not exorbitant, while being small enough to capture all possible variations in RSS measurements between adjacent samples. Therefore, an appropriate strategy is to expect exactly one sensor to exist in the non-overlapping area of two circles centered at two adjacent sampled points. In Fig. 4.9, this means that there is one sensor in the shaded area, which results in the following condition for choosing Δ:

$$2\rho[(\pi - 2\phi)r^2 + r\Delta \sin\phi] = 1, \tag{4.16}$$

where $\phi = \arccos[\Delta/(2r)]$.

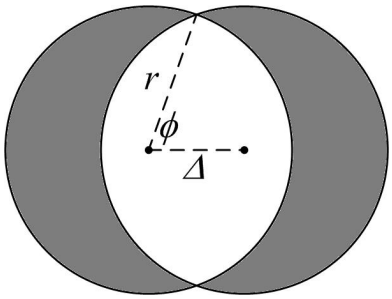


Figure 4.9: Illustration of calculating the sample interval.

4.4.3 The Special Case of Out-of-range Primary Users

When an estimated location of a primary signal transmitter appears close to the border of the WSN, it is possible that the transmitter is a legitimate user located out of the WSN[3]. Given this special case, it is necessary to distinguish whether a detected transmitter is a PUE attacker located on the border of the WSN or it is a transmitter of a primary user that is located out of the range of the WSN. We assume that a primary user's location is known ahead of time, since only TV systems are considered and the location of a TV tower is public information. Then we develop the following approach to compute the likelihood that a detected signal is coming from the primary user's location and from the border of the WSN. By comparing the likelihoods of the two events, one can derive the transmitter's location.

Assume that one transmitter's position derived in Subsection 4.4.2 to be (X_1, Y_1). When (X_1, Y_1) is located close to the border of the deployed WSN, we want to know whether it is more probable that the transmitter is in fact at a known position (X_2, Y_2) that is out of the range of the WSN. Assume that the RSS measurement in the WSN has been smoothed by taking the median of the RSS values within a circular region of radius r. We randomly sample K smoothed measurements across the WSN, with each measurement corresponding to a location (x_k, y_k) and a smoothed RSS value

[3]Note that since a PUE attacker transmits at relatively low transmission power, the attacker has been assumed to be always within the range of the CR network and its underlying WSN so that its attack remains effective.

R_k (in dBm) from n_k sensors in the region, where $k = 1, \ldots, K$. As discussed before, since a reasonable sampling space is Δ, when all sampling possibilities are considered, the maximum K will be $D_x D_y/\Delta^2$. Then a two-step process is executed to calculate the likelihood that the transmitter is from (X_h, Y_h), where $h = 1, 2$. In the first step, a linear optimization operation is executed to make an estimation of transmission power p_h, in which the difference between the smoothed RSS values and what are predicted by the log-loss signal propagation model is minimized.

$$\begin{aligned} &\min \sum_{k=1}^{K} (u_k + o_k) \\ &\text{s.t.} \forall k = 1, ..., K : \\ &R_k + u_k - o_k = p_h + \beta_0 + \beta_1 \ln \sqrt{(X_h - x_k)^2 + (Y_h - y_k)^2} \\ &u_k, o_k \geq 0. \end{aligned} \tag{4.17}$$

The variables u_k and o_k both represent the absolute difference between R_k and the value predicted by the model. The formulation of the above linear optimization problem mandates that when R_k is greater than what is predicted by the model, u_k is zero and o_k is the difference. When R_k is smaller, o_k is zero and u_k is the difference. The solution to (4.17) generates an estimated p_h. With the knowledge of p_h, the normalized difference for the scenario that the transmitter is located at (x_h, y_h) is computed as

$$D_h = \frac{1}{K} \sum_{k=1}^{K} \left| R_k - p_h + \beta_0 + \beta_1 \ln \sqrt{(X_h - x_k)^2 + (Y_h - y_k)^2} \right| \frac{\sqrt{n_k}}{\sigma}. \tag{4.18}$$

When (X_h, Y_h) is indeed the transmitter's location, the expected value of each item in the summation should approach zero. In contrast, if (X_h, Y_h) is not the transmitter's location, each item in the summation will deviate from zero. Therefore, D_h can be used to compare the likelihoods that the transmitter is at specific locations—i.e., the location of the transmitter is decided to be (X_{h_0}, Y_{h_0}), where

$$h_0 = \arg\min_h D_h. \tag{4.19}$$

4.4.4 Security Analysis

In this subsection, we explore the security aspects of the proposed localization system in a hostile environment. In particular, we consider two categories of potential attacks and analyze their impacts.

The first category of attacks aim to escape localization by disrupting RSS measurements. Attackers may manipulate their signal transmission either temporally or spatially. In temporal manipulation, an attacker may take either of the following two approaches. With the first approach, the attacker may vary its transmission power over time in an attempt to cause confusion. However, this attack has limited impact since the proposed localization scheme collects and analyzes a snapshot of RSS measurement, in which only one transmission power value is in effect. With the second approach, the attacker may temporarily stop transmission when it knows that a snapshot of RSS measurement is being taken. However, RSS measurement is not only used for localization, but more importantly, it is the premise of spectrum sensing. To successfully launch a PUE attack, an attacker's signal has to be detected in the spectrum sensing process. Therefore, an attacker cannot benefit from keeping silent while RSS measurements are being collected.

An attacker has two options to conduct spatial manipulation of its transmission. As the first option, the attacker can install a directional antenna so that it is detected by less number of sensors. Because the attacker's signal is still detected by some sensors, the effect of its PUE attack remains unchanged. On the other hand, less RSS information will lead to vaguer peak locations in an RSS snapshot, thereby adding difficulty to localization. In Section 4.5, we will further investigate this problem using simulation. The second method for spatial manipulation is to use multiple transmitters deployed at different locations. Because the signals emitted by the transmitters interfere with each other, the signal characteristics (e.g., time of arrival, angel of arrival, RSS) of different transmitters may be mixed together, causing wrong localization results. However, the proposed

localization system is able to identify multiple transmitters if multiple RSS peaks are observed. In Section 4.5, its performance will be evaluated when multiple transmitters are present.

The second category of attacks disrupt localization by injecting false data to the localization system. This is possible when some sensor nodes are compromised. This attack is partly mitigated by the fact that the median value has been used for RSS smoothing. It is known that in the absence of noise, taking the median can tolerate up to 50 percent outliers among all measurements [60].

4.5 Simulation of the Localization Scheme

4.5.1 Simulation Settings and Objectives

In the simulation, a 2000m$\times$2000m CR network with an underlying WSN of the same size was assumed and the statistical log-loss propagation model with (β_0, β_1, σ) = (-30.00, -10.00, 10.0) was used. The exact values of these parameters are unknown to the localization system, but we assume that they are estimated using the offsite calibration scheme proposed in [61], where a realistic estimation was given as (β_0, β_1, σ) = (-32.03, -9.73, 10.0). Then based on (4.15), assuming $m = 3$dB and $P = 0.9$, we generated seven simulation settings representing various density of sensors in the WSN and their corresponding parameters r and d, which are shown in Table 4.1. We used $\Delta = r/15$ for the simulation so that the condition in (4.16) is satisfied as well. We consider four cases when there is a single transmitter, when an attacker uses a directional antenna, when multiple PUE attackers exist, and when it is the special case of out-of-range primary users.

We evaluate the system's localization error and computation time. Based on the discussion in Section 4.1, the metric of localization error has the following meaning. When a primary signal

Table 4.1: Simulation settings of the non-interactive localization scheme.

Sensor density (m^{-2})	Number of sensors	r (m)	d (m)
2.5×10^{-5}	100	305	1,294
5×10^{-5}	200	300	1,273
1.25×10^{-4}	500	200	849
2.5×10^{-4}	1,000	200	849
5×10^{-3}	2,000	100	424
1.25×10^{-4}	5,000	100	424
2.5×10^{-3}	10,000	50	212

transmitter is found to be away from any known location of primary users more than the localization error, the transmitter is deemed as a PUE attacker. Once a PUE attacker is detected, the localization error defines a range of area for pinpointing the attacker. The computation time is the time to run the localization algorithm but does not include the WSN's network delay for collecting data. The computation time shows the relative computation overhead in different scenarios. It is measured in our specific simulation environment and its absolute value could change as the environment varies[4].

Each simulation run simulates a transient process in which one or more primary signal transmitters transmit signals that are received by the WSN. Then based on the RSS distribution collected from the WSN (suppose the RSS raw data are collected correctly), the localization scheme proposed in 4.4 immediately runs and outputs transmitter locations. For every simulation, we carry out at least ten simulation runs to generate each datum. Some simulation requires more runs to get a stable result. We will state the exact number of simulation runs clearly in individual simulations presented in 4.5.2. Wherever possible, we also plot an error bar for each datum to show the range of possible simulation results.

[4]In particular, the simulation was run in MATLAB on a P4 2.8GHz, 512M RAM PC.

4.5.2 Simulation Results

4.5.2.1 Localization Error of a Single Transmitter

We consider three scenarios, in which a 500mW primary signal transmitter is in the center, on the border, and on the corner of the WSN, i.e., T_1 at (1000m, 1000m), T_2 at (1000m, 50m), and T_3 at (50m, 50m), respectively. The localization errors of the proposed localization system under various settings are shown in Fig. 4.10. In the figure, every datum is the average of ten independent simulations. The results prove the localization system to be effective. For example, under the 10,000-sensor scenario, the expected space of two adjacent sensors is 20m, which is close to the localization error of T_1, i.e., 21.9m. T_2 and T_3 have relatively greater localization error because on the border or on the corner of the WSN, there are less number of sensors around, resulting in less measurements and thus poorer accuracy. Meanwhile, the computation time is shown to be affordable. T_2 and T_3 require relatively greater computation time because less number of measurements means more ambiguity and causes the localization algorithm to sample greater number of regions (i.e, the set R has more elements).

4.5.2.2 The Case of Directional Antenna

An attacker may mount a directional antenna to evade localization. To investigate its impact, we repeated the previous simulation assuming that the primary signal transmitter used a ten-element Yagi-Uda antenna. A ten-element Yagi-Uda antenna is a typical directional antenna and its radiation pattern can be found in [59]. In the simulation, the major lobe in the antenna's radiation pattern pointed to the east. As the results in Fig. 4.11 show, the directional antenna has increased the localization error. We reason that the use of directional antennas caused less number of sensors to detect the transmitted signals and this had the same effect as decreasing the density of the

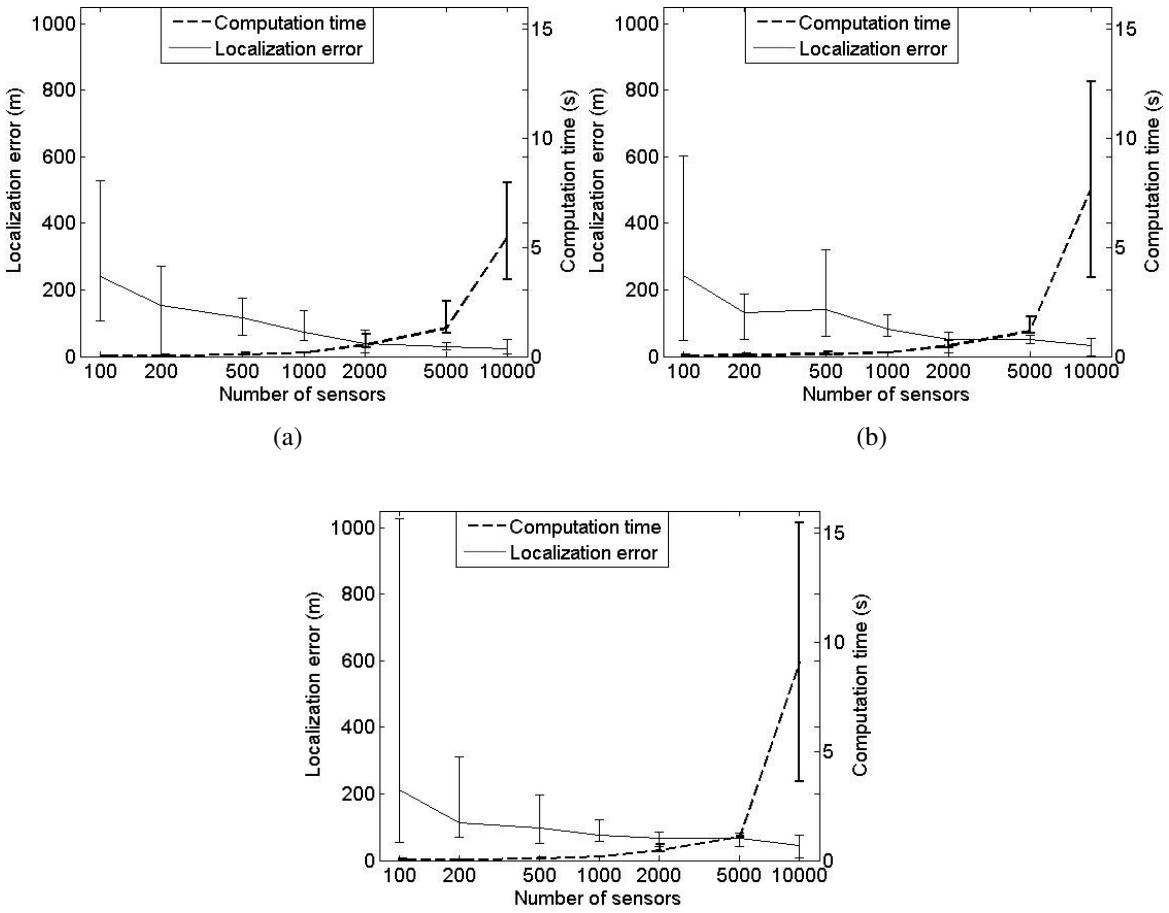


Figure 4.10: The localization error of the proposed localization system. (a) T_1(1000m, 1000m). (b) T_2(1000m, 50m). (c) $T_3(50m, 50m)$.

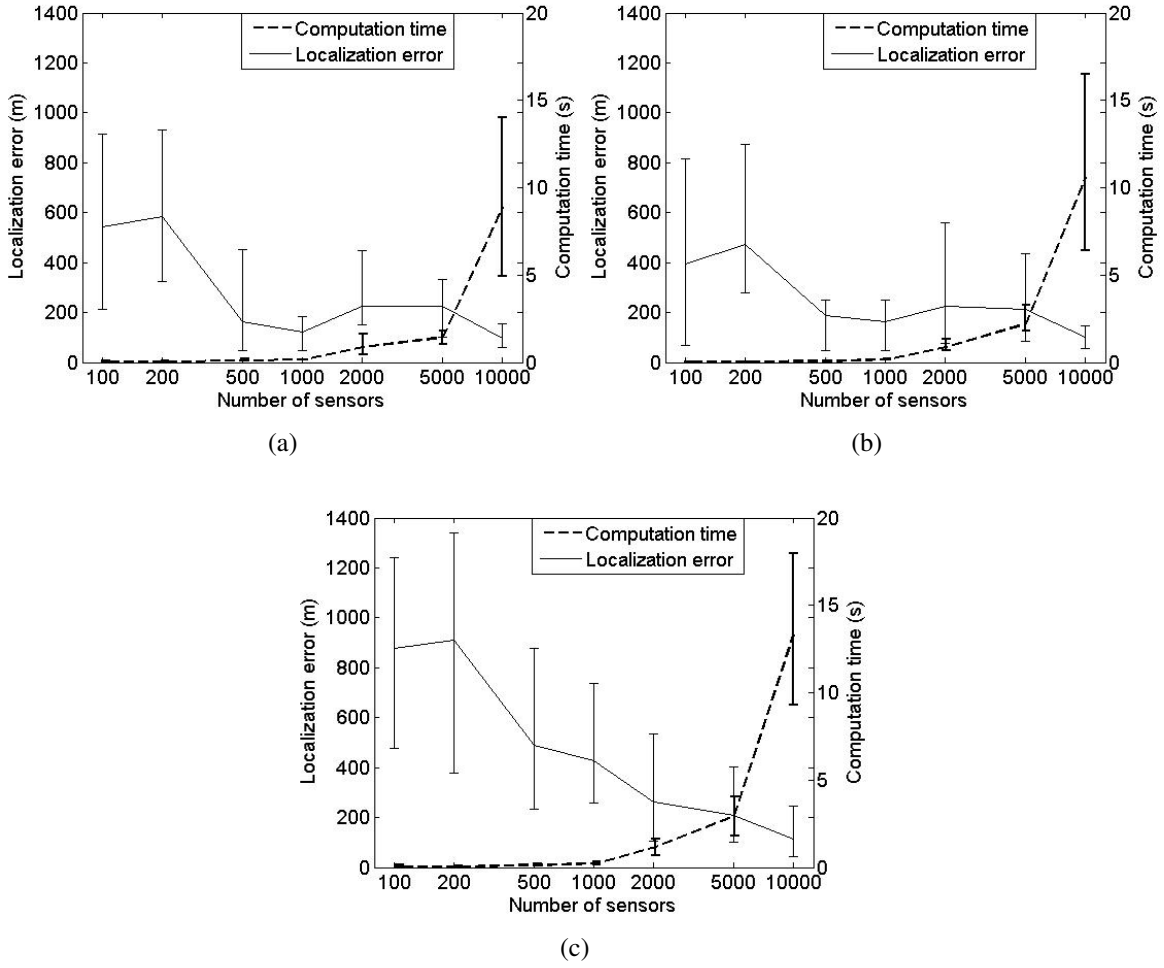


Figure 4.11: The system's localization error when a primary signal transmitter uses a ten-element Yagi antenna. (a) T_1(1000m, 1000m). (b) T_2(1000m, 50m). (c) $T_3(50m, 50m)$.

sensors.

Another observation is that for locations T_1 and T_2, the localization errors for 500-sensor and 1,000-sensor scenarios are smaller than those for 2,000-sensor and 5,000-sensor scenarios, which is counter-intuitive. Further research revealed that because the directional antenna brought about the equivalent effect of decreasing the sensor density ρ, the derived values of r and d using (4.15) became too small, causing the localization algorithm to be trapped at a local maximum. To confirm this reasoning, we doubled the values of r and d and repeated the previous simulation. Fig. 4.12

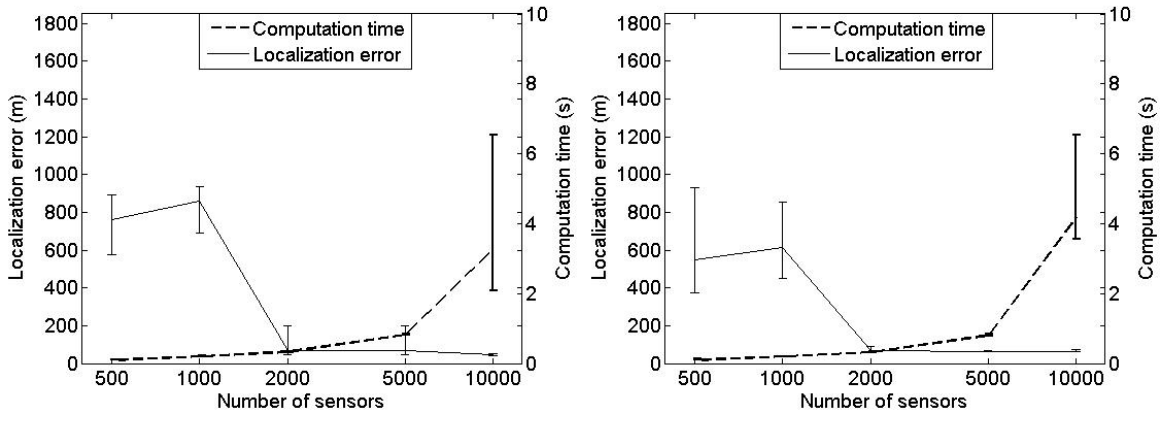
shows that with this change, for high-sensor-density scenarios, the performance was greatly improved. However, for other scenarios with low sensor densities, this caused overly large region sizes and the localization error was significantly increased.

4.5.2.3 The Case of Multiple PUE Attackers

A two-transmitter scenario is considered under 2,000-sensor and 5,000-sensor deployments. Both are assumed to be transmitting signals at the same UHF 617MHz band and their phase shift was randomly chosen. We varied the distance between the two transmitters and observed the estimated number of transmitters and their locations by the localization system. Based on 100 independent simulation runs, Fig. 4.13 shows the ratio of the runs that output correct number of transmitters, i.e., two. Fig. 4.14 shows the corresponding localization errors when the number of transmitters was correctly recognized. When the two transmitters are within 500 meters of each other, the localization system only recognizes one signal source most of the time. However, when the distance increases, the two transmitters can be both correctly localized, with a localization error similar to that in single-transmitter scenarios.

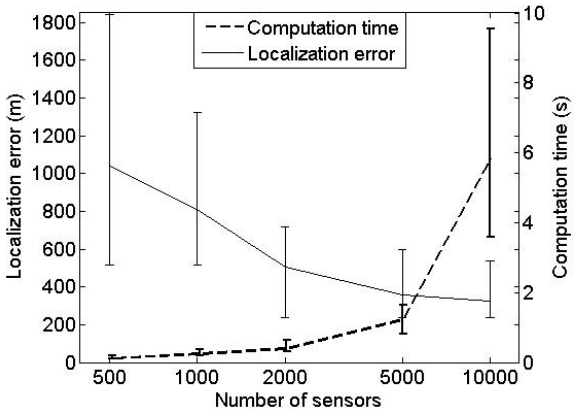
4.5.2.4 The Case of Out-of-range Primary Users

In Subsection IV.D, the value of D_h was used to compare the likelihoods that a primary signal transmitter is on the border of the WSN and in out-of-range locations. We fixed a PUE attacker at location (1950m, 1000m) and set a primary user at location (($1950 + \delta_x$)m, 1000m), where δ_x is a variable in the simulation. The PUE attacker is transmitting while the primary user is not transmitting. The result in Fig. 4.15 shows that when the distance between the PUE attacker and the out-of-range primary user is large, the D_h value induced by the attacker is much smaller than that induced by the primary user, showing that the attacker's location will be correctly output by



(a)

(b)



(c)

Figure 4.12: The system's localization error when r is doubled in case that a primary signal transmitter users a ten-element Yagi antenna. (a) T_1(1000m, 1000m). (b) T_2(1000m, 50m). (c) T_3(50m, 50m).

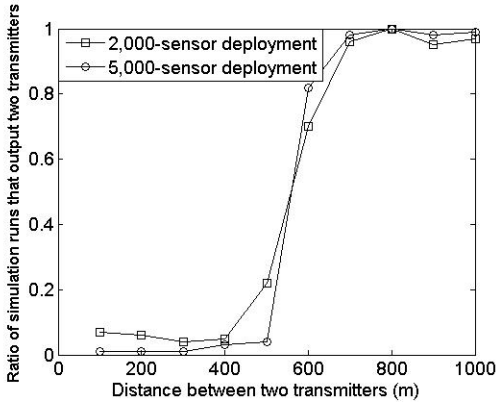


Figure 4.13: The ratio of simulation runs that correctly recognize the number of transmitters in a two-transmitter scenario.

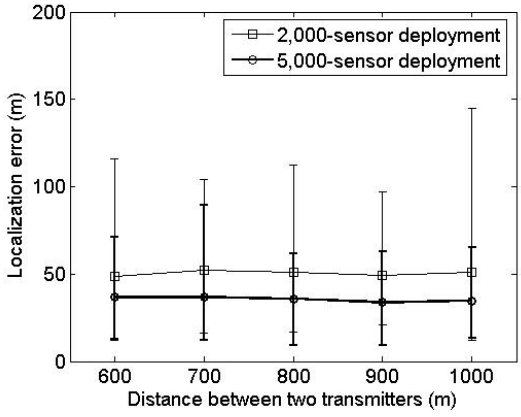


Figure 4.14: The localization error in a two-transmitter scenario.

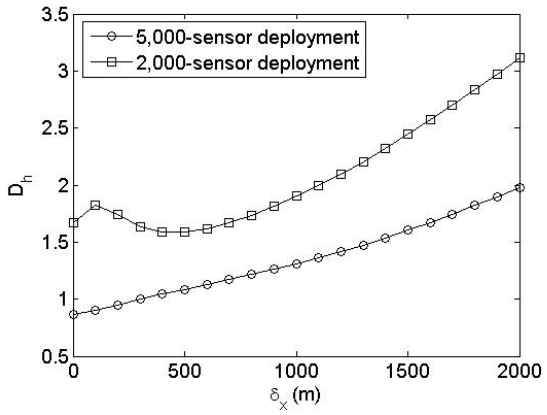


Figure 4.15: D_h vs. δ_x.

(4.19). However, when the distance between the PUE attacker and the primary user is relatively small, due to the modeling error and the localization error, D_h cannot be used for distinguishing the attacker from the primary user. In this case, as the flowchart in Fig. 4.1 shows, the signal energy level will be further examined to judge the legitimacy of the transmitter. When the attacker and the primary user are located close to each other, the difference between their signal energy levels levels will be distinguishing.

4.6 Chapter Summary

This chapter has presented a transmitter verification scheme to detect PUE attacks in CR networks. One of the distinguishing features of the proposed transmitter verification scheme is the fact that it uses the transmitter's position in the verification process. The core of the transmitter verification scheme is a non-interactive location verification scheme or a non-interactive localization scheme. For the location verification scheme, this chapter proposed DRT and DDT. DRT is based on the RSS of a signal source while DDT utilizes the received signal's relative phase difference when the signal is received at different receivers. Simulation results show that several factors, such as the

location of the attacker's transmitter relative to the LVs, can impact the performance of the two location verification schemes. For the localization scheme, this chapter proposed a localization system that uses an underlying WSN to collect RSS distributions in a CR network and uses data smoothing to pinpoint a transmitter's location. Security analysis and simulation results show that the proposed localization scheme is effective and robust against a number of attacks.

Chapter 5

A Novel Data Fusion Technique Robust against SSDF Attacks

As discussed in 3.2, SSDF attacks are a major threat to the data fusion process of the DSS in a CR network. To maintain an adequate level of accuracy in the midst of SSDF attacks, the data fusion technique used in DSS needs to be robust against fraudulent local spectrum sensing results reported by malicious secondary users. However, the fusion techniques in 3.2 have not considered SSDF attacks and they are inadequate to ensure the robustness of the final sensing decision. They share two common properties that contribute to their vulnerability to SSDF attacks. First, none of these techniques can guarantee both a bounded false alarm probability and a bounded miss detection probability. In an adversarial environment, these probabilities will get larger. Second, these techniques treat all sensing terminals indiscriminately, regardless of whether a sensor is reporting data correctly or incorrectly.

This chapter considers the defense against SSDF attack from the perspective of data fusion. We propose a new technique to improve robustness against SSDF attacks. This chapter is organized as

follows. Section 5.1 discusses the method to apply the Sequential Probability Ratio Test (SPRT) to the data fusion in DSS. Section 5.2 presents a weighted SPRT technique, which is a reputation-based data fusion scheme that improves SPRT's robustness against SSDF attacks. In Section 5.3, we quantitatively compares existing fusion techniques using extensive simulation results. Then in Section 5.4, practical considerations on the application of various fusion techniques are discussed. The chapter ends with a summary in Section 5.5.

5.1 Sequential Probability Ratio Test (SPRT)

One method of increasing the robustness of the data fusion process is to employ SPRT, which is a data fusion scheme that supports sampling a variable number of local spectrum sensing reports [71]. SPRT has the desirable property that guarantees both a bounded false alarm probability and a bounded miss detection probability in a non-adversarial environment. This is an advantage over the techniques discussed previously.

To apply SPRT to data fusion for DSS, one needs to define the following likelihood ratio as the decision variable:

$$S_n = \prod_{i=0}^{n} \frac{P[u_i|H_1]}{P[u_i|H_0]}. \tag{5.1}$$

Note that the number of samples n is a variable and can be different from $m+1$ (see Fig. 3.4).

The fusion decision is based on the following criteria:

$$\begin{cases} S_n \geq \eta_1 \Rightarrow \text{accept } H_1, \\ S_n \leq \eta_0 \Rightarrow \text{accept } H_0, \\ \eta_0 < S_n < \eta_1 \Rightarrow \text{take another observation.} \end{cases} \tag{5.2}$$

The values of η_1 and η_0 are decided by

$$\eta_1 = \frac{1-P_{01}}{P_{10}} \text{ and } \eta_0 = \frac{P_{01}}{1-P_{10}},$$

where P_{01} and P_{10} are the tolerated false alarm probability and the tolerated miss detection probability, respectively. Assuming K is the expected value of n to accept hypothesis H_1 or H_0, it can be proved that K is minimized in SPRT given the value of P_{01} and P_{10} [71].

As the above discussion shows, SPRT requires the same knowledge of *a priori* probabilities that Bayesian detection and Neyman-Pearson Test require, i.e., $P(u_i|H_1)$ and $P(u_i|H_0)$. Existing research assumes that these probabilities exist as empirical data [40]. However, in practice such data may not be available. Even if such data are available, because *a priori* probabilities change with a sensing terminal's location, empirical data would need to be re-collected every time the sensing terminal moves to a different location. Here we propose an approach to calculate the probabilities based on the log-normal shadowing path loss model [57]. The advantage of this approach is that the calculation method utilizes the physical location of a sensing terminal. Thus, when a sensing terminal moves to a different location, *a priori* probabilities can be immediately calculated without waiting to collect new empirical data. The log-normal shadowing path loss model can be represented as:

$$PL(d) = \overline{PL}(d) + X_\sigma = \overline{PL}(d_0) + 10l\log(\frac{d}{d_0}) + X_\sigma, \tag{5.3}$$

where d is the transmitter-receiver distance, $PL(d)$ is the path loss as a function of d, $\overline{PL}(d)$ is the mean of $PL(d)$, X_σ is a zero-mean Gaussian distributed random variable with standard deviation σ, d_0 is a close-in reference distance which is determined from measurements close to the transmitter, and l is the path loss exponent which indicates the rate at which the path loss increases with distance. All items in the equation are in dB.

The received power $P_r = P_t - PL(d)$, where P_t is the transmitted power, and both P_r and P_t are in dB. Assuming the receiver uses an energy detector with a detection threshold γ, the *a priori*

probabilities under H_1 can be computed as:

$$\begin{aligned} P(u_i = 1|H_1) &= P(P_r > \gamma|H_1) \\ &= P(X_\sigma < P_t - \overline{PL}(d) - \gamma) \\ &= Q\left(\frac{\gamma - P_t + \overline{PL}(d)}{\sigma}\right) \end{aligned} \tag{5.4}$$

$$\begin{aligned} P(u_i = 0|H_1) &= 1 - P(u_i = 1|H_1) \\ &= Q\left(\frac{P_t - \gamma - \overline{PL}(d)}{\sigma}\right) \end{aligned} \tag{5.5}$$

In the above derivations, $(P_r > \gamma)$ represents the condition that an energy detector detects a received signal, P_r is expanded using $P_t - PL(d)$, and $PL(d)$ is further expanded using (5.3).

When hypothesis H_0 holds, $P_r = n_0$, where n_0 can be regarded as a Gaussian noise power with mean $\bar{n}_0$ and standard deviation σ_n. Similarly the *a priori* probabilities under H_0 can be computed as:

$$P(u_i = 1|H_0) = Q\left(\frac{\gamma - \bar{n}_0}{\sigma_n}\right) \tag{5.6}$$

$$P(u_i = 0|H_0) = Q\left(\frac{\bar{n}_0 - \gamma}{\sigma_n}\right) \tag{5.7}$$

It should be noted that the function $\overline{PL}(d)$ and the parameter $\bar{n}_0$ need only to be calibrated for the type of environment (rural, urban, etc.) in question. As long as a secondary user moves within the environment, they do not have to be re-calibrated for a different location. In contrast, with empirical *a priori* probabilities at a location, when a secondary user moves, one has to re-calibrate the probabilities using empirical data at the new location. Therefore, the proposed approach is more flexible for practical use.

5.2 Weighted Sequential Probability Ratio Test (WSPRT)

When SSDF attacks are considered, SPRT still shares the same drawback with previous schemes in that all sensing terminals are treated indiscriminatingly, regardless of whether a sensing terminal is

sending reliable or unreliable local spectrum sensing reports. Therefore, in order to improve SPRT, we propose a reputation-based scheme called WSPRT.

WSPRT is composed of two steps. The first step is a reputation maintenance step, and the second step is the actual hypothesis test. In the reputation maintenance step, a sensing terminal's reputation ratings are allocated based on the accuracy of a sensing terminal's sensing report in relationship to the final sensing decision: the reputation value is set to zero at the beginning; whenever its local spectrum sensing report is consistent with the final sensing decision, its reputation is incremented by one; otherwise it is decremented by one. Under this rule, assuming N_i's reputation value is r_i, the last sensing report N_i sends to N_0 is u_i, and the final decision is u, then r_i is updated by: $r_i \leftarrow r_i + (-1)^{u_i+u}$. It can be proved that given that the final decision is true at a probability greater than 0.5, a sensing terminal with more accurate local sensing report has a higher expected reputation value than a terminal with less accurate sensing report.

Next the reputation value should be used for the hypothesis test. We define the weight of N_i as w_i, which is a function of r_i:

$$w_i = f(r_i) \tag{5.8}$$

Our objective is to modify the likelihood ratio in (5.1) by adding a component w_i so that the decision variable also takes a sensing terminal's reputation into consideration. The proposed new decision variable is

$$W_n = \prod_{i=0}^{n} \left(\frac{P[u_i|H_1]}{P[u_i|H_0]} \right)^{w_i} \tag{5.9}$$

In order for such a decision variable to be robust against SSDF attacks, $f(\cdot)$ should satisfy two requirements:

1. $f(\cdot)$ accepts an r_i with arbitrary value and outputs a $w_i \in [0, 1]$, and $f(\cdot)$ should be an non-decreasing function of r_i, i.e., $w_i \geq w_j$ if $r_i \geq r_j (i, j = 0, \ldots, m; i \neq j)$. Also,

$f(\max(i)) = 1$ and $\lim_{r_i \to -\infty} f(r_i) = 0$. This requirement ensures that a sensing report from the highest reputation sensing terminal is fully trusted—(5.9) degenerates to (5.1) if $w_i = 1$. On the other hand a sensing report from a very low reputation sensing terminal is ignored—the multiplier in (5.9) becomes one and does not change the decision variable when $w_i = 0$. Also, a non-decreasing $f(\cdot)$ ensures that a sensing report from a sensing terminal with higher reputation always has a higher weight in changing the decision variable than a sensing report from a sensing terminal with lower reputation does.

2. $f(\cdot)$ should ensure that enough weight is allocated to a sensing terminal that has a slightly negative reputation value. This requirement is necessary because at the beginning of a WSPRT process, a "good" sensing terminal (i.e., sending correct sensing reports for the most of time) may send incorrect sensing reports due to randomness and get a slightly negative reputation value. If the weight for such a sensing terminal is too small, the sensing reports from malicious sensing terminals will play more important roles in the final sensing decision. Because the final sensing decision is the criteria to maintain sensing terminals' reputation, a wrong sensing decision has a cumulative effect on subsequent sensing decisions. However, if we allow the reputation maintenance to run a few steps without significantly decreasing the weight for sensing terminals that send wrong reports, then the sensing decisions will be less subject to short-term randomness. Thus, when reputation ratings are accumulated for a few steps, a single wrong sensing decision will no longer have much impact on the overall reputation values, and the cumulative effect on subsequent sensing decisions will also be limited.

Based on the above requirements, we use the following function for $f(\cdot)$:

$$w_i = f(r_i) = \begin{cases} 0 & r_i \leq -g \\ \frac{r_i + g}{\max(r_i) + g} & r_i > -g \end{cases}.$$

The variable $g(>0)$ is used to meet the second requirement. In particular, w_i for a good sensing terminal will not be zero for the first $(g-1)$ reputation maintenance steps. For the g-th reputation maintenance step, $P[r_i \leq -g] < 2^{-g}$. This probability is very small if g is assigned a relatively small number. (Note that g should be small enough to let the reputation scheme be sufficiently sensitive to incorrect sensing reports.) For example, when $g = 5$, the probability is less than 0.03125. Therefore, it can prevent allocating overly small weight for a good sensing terminal that has a slightly negative reputation value at the beginning.

Given all the components discussed above, the data fusion via WSPRT can be summarized as the following algorithm.

1: $\forall i, r_i = 0$.

2: For each spectrum sensing attempt made by N_0 {

3: $i = 0$, $W_n = 1$.

4: Get a spectrum sensing report u_i from N_i.

5: $W_n \leftarrow W_n \cdot \left(\frac{P[u_i|H_1]}{P[u_i|H_0]}\right)^{f(r_i)}$.

6: If $\eta_0 < W_n < \eta_1$, $i \leftarrow (i+1) \bmod (m+1)$. Go to step 4.

7: If $W_n \geq \eta_1$, accept H_1, i.e., output $u = 1$. Go to step 9.

8: If $W_n \leq \eta_0$, accept H_0, i.e., output $u = 0$.

9: For each sampled u_i, set $r_i \leftarrow r_i + (-1)^{u_i+u}$.

10:}

5.3 Simulations

5.3.1 Simulation Setup

We carried out simulations in MATLAB to test and compare all the previously discussed data fusion schemes. The simulation implements the path loss model in 5.3. Under the path loss model, we investigate how well each data fusion scheme outputs a final spectrum sensing result. We consider the data fusion schemes' performance under varying attack types and strength, varying incumbent signal strength, and varying node density.

In the simulations, N secondary users are randomly located in a 2000m$\times$2000m ad hoc CR network area, each with a transmission range of 250m. As discussed in 3.1, this range value is consistent with the protocol interference model [31]. Among the N secondary users, there are N_a SSDF attackers. We consider two types of SSDF attacks: always-false and always-free. An always-false attacker always sends sensing reports that are opposite to its local spectrum sensing results while an always-free attacker always reports spectrum to be fallow. Each secondary user moves according to the random waypoint mobility model (see 3.1) within the range of the network area. Each node moves with a maximum speed of 10m/s, a minimum speed of 5m/s, and a maximum idle time of 120s. Similar to the simulations conducted in 3.1, we simulated one hour before the real simulation to ensure that the random waypoint model entered steady state. The primary user, a TV tower with a duty cycle of 0.2, is located D meters away from the center of the CR network. See Fig. 5.1 for the simulation network model.

The $\overline{PL}(d)$ in (5.3) employs the HATA model [57], which has been suggested by the 802.22 working group as the path loss model for a representative CR network environment [14]. The HATA model has different versions for urban and rural environments. We used the one for rural environments since the real implementation of CR networks is likely to first occur in rural areas

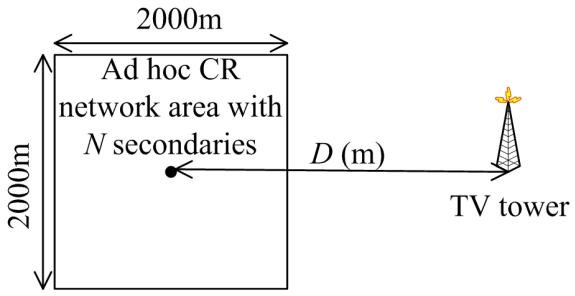


Figure 5.1: Simulation layout.

where licensed spectrum is less utilized. The model is given by

$$\begin{aligned}\overline{PL}(d) = \; & 27.77 + 9.39\log f_c - 4.78(\log f_c)^2 - 13.82\log h_{te} \\ & -(1.1\log f_c - 0.7)h_{re} + (44.9 - 6.55\log h_{te})\log d\end{aligned} \tag{5.10}$$

where f_c is the signal frequency, h_{te} is the effective transmitter antenna height in meters, and h_{re} is the effective receiver antenna height in meters, and d is the transmitter-receiver distance in kilometers. All items in (5.10) are in dB. In our simulation, we assume that the primary user works at the UHF frequency of 617MHz, $h_{te} = 100$m, and $h_{re} = 1$m, respectively. At the incumbent transmitter side, the EIRP is assumed to be 100kW. At the secondary receiver side, a simple energy detector is assumed. Each receiver has a typical sensitivity of -94dbm, which is the minimum power for a signal to be detected. For the noise power, the typical value of $\bar{n}_0$ in the considered band frequency -106dBm is used. For the deviation part of the log-normal shadowing path loss model and noise, we adopted $\sigma = \sigma_n = 11.8$, whose values were reported in [63].

A secondary user acts as both a sensing terminal and a data collector. DSS at each secondary user is periodically repeated at an interval of 30s, and each simulation lasts for two hours. This is equivalent to 240 iterations of DSS for each node. Because in most simulations there are 500 nodes in a CR network, there are totally 120,000 data fusion operations in these simulations. We have found very stable statistical results given the number of iterations.

We simulated and compared eight different data fusion techniques. For decision fusion techniques,

the three variants of AND, OR, and Majority rules are simulated. For Bayesian detection and Neyman-Pearson test, since they both boil down to a fixed-number likelihood ratio test (LRT) with different thresholds, we simulate the two techniques together under the name of "LRT" and use three different thresholds for them. The first threshold is calculated from the right hand side of (3.1) by assuming the perfect knowledge of P_0 and P_1, i.e., $P_0 = 0.8$ and $P_1 = 0.2$. The costs are assigned as: $C_{00} = C_{11} = 0$, $C_{10} = 1$, and $C_{01} = 10$, which were also the cost assignments used in [40]. With these values, the first threshold can be calculated as $\lambda_1 = 0.4$. Because the accurate knowledge on P_0 or P_1 may not be available in practice, we simulated two other thresholds $\lambda_2 = 4\lambda_1$, $\lambda_3 = \lambda_1/4$. One can also understand the two thresholds as the result of adopting different strategies for cost assignments. They are equivalent to assigning C_{01} with the values 40 and 2.5, respectively. Another two simulated fusion techniques are SPRT and WSPRT. The parameters in these two fusion techniques used in the simulation are $P_{01} = 10^{-5}$, $P_{10} = 10^{-6}$, and $g_i = 5$. The selection of the first two parameters aims to guarantee small false alarm and miss detection probabilities and the selection of g_i has been discussed in 5.2.

5.3.2 Simulation Results

5.3.2.1 Objectives

We are interested in four metrics: false alarm ratio, miss detection ratio, correct sensing ratio, and number of samples. The first two metrics have been discussed before. The correct sensing ratio is the number of correct final sensing decisions divided by the number of total sensing decisions. Therefore, the first three metrics should add up to one. The number of samples refers to the average number of samples a secondary user needs to collect from each neighbor to make a final decision,

and it measures the overhead of a particular data fusion technique. For decision fusion and fixed-number likelihood ratio test, the number of samples is always one. Only for SPRT and WSPRT the number of samples changes. Therefore, we study this metric only for SPRT and WSPRT.

5.3.2.2 Impact of Varying Attack Strength

In this set of simulations, we fix $N = 500$ and $D = 3000$, while changing attack types and varying N_a from 0 to 100 at an interval of five. Figs. 5.2 and 5.3 show the simulation results when we consider always-false and always-free attacks, respectively. In all cases, the decision fusion with an "OR" rule has the lowest miss detection ratio and the highest false alarm ratio. In contrast, the decision fusion with an "AND" rule has the highest miss detection ratio and the lowest false alarm ratio. These two techniques are not favorable, though, since they end up with either almost always outputting "occupied" or almost always outputting "fallow". For all other fusion techniques, when there is no attacker, they all can effectively generate accurate sensing decisions—the overall correct sensing ratios are all above 96%. However, their performances diverge when SSDF attacks are introduced. When always-false SSDF attacks are introduced, Fig. 5.2 shows that the false alarm ratios and miss detection ratios for decision fusion with a "Majority" rule, SPRT, WSPRT, and LRTs are all increased. Among these techniques, SPRT experiences the greatest magnitude of increase in terms of the two ratios, which shows that SPRT is the least robust against always-false SSDF attacks. We understand it as the fact that SPRT does not consider attacks and it may collect multiple reports from malicious secondary users, which amplifies the effect of attacks. In contrast, WSPRT is shown to be the most robust against always-false SSDF attacks and maintains a correct sensing ratio as high as 96.8% when there are 100 attackers. This shows that the weight scheme has taken effect. However, the better correct sensing ratio comes with a cost—the number of samples has been increased to 4.5-5.5 times for WSPRT, and the more the attackers are, the more samples it

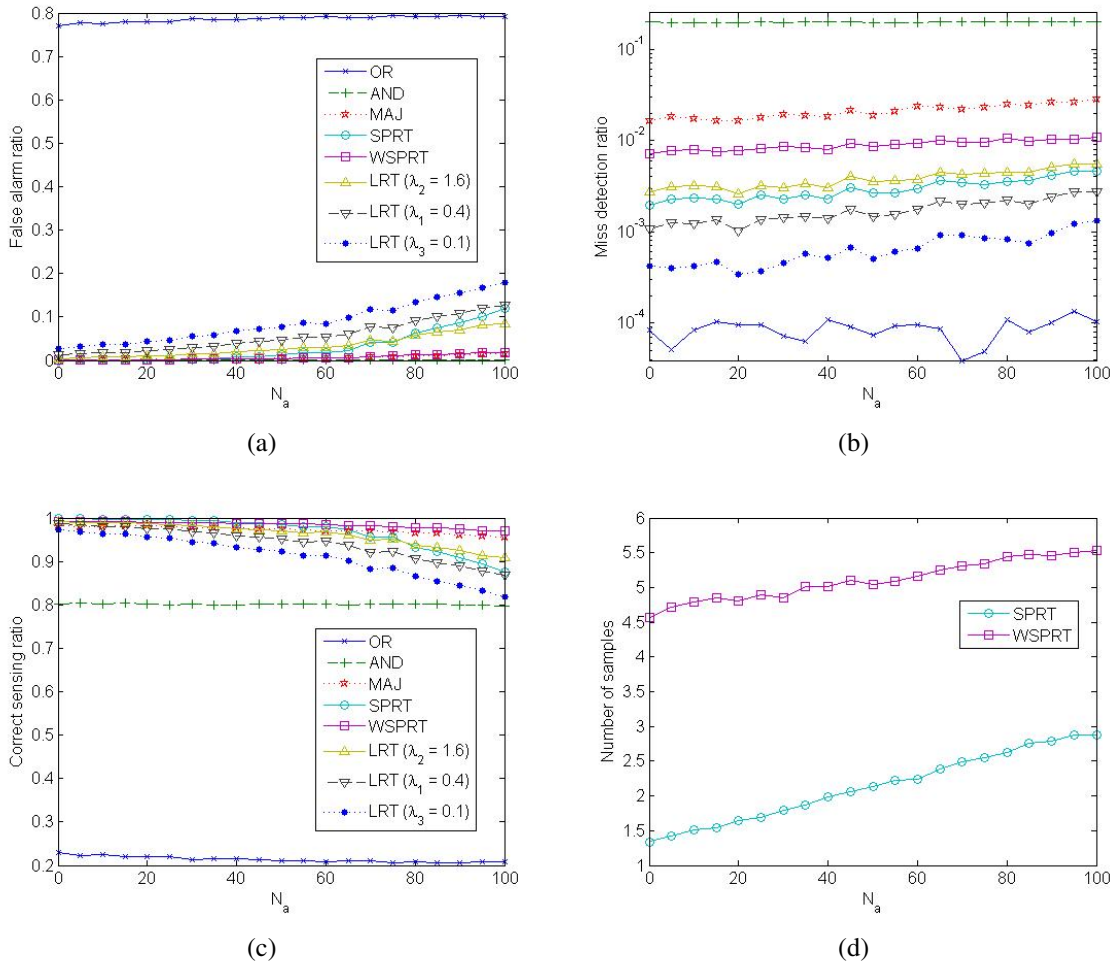


Figure 5.2: The performance of eight fusion techniques when the number of always-false SSDF attackers changes: (a) false alarm ratio, (b) miss detection ratio, (c) correct sensing ratio, and (d) number of samples.

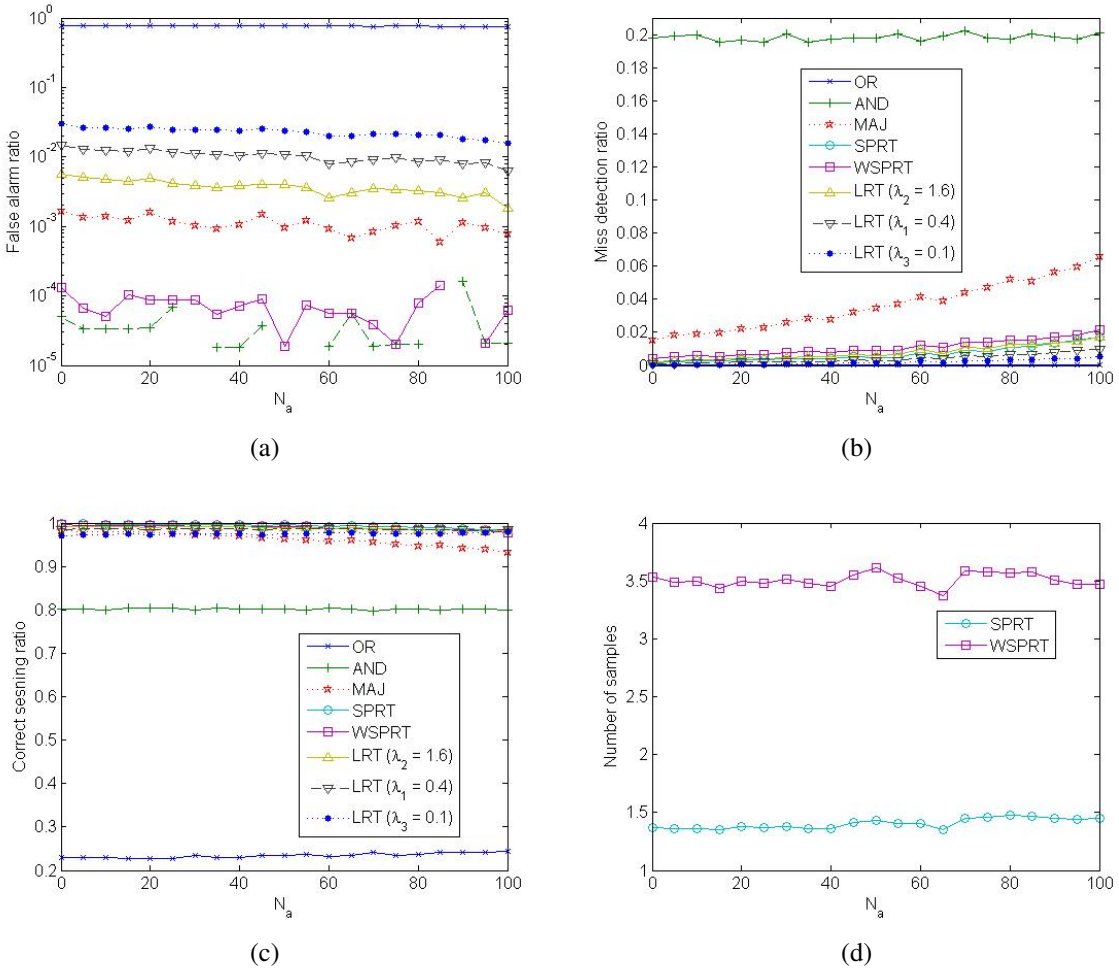


Figure 5.3: The performance of eight fusion techniques when the number of always-free SSDF attackers changes: (a) false alarm ratio, (b) miss detection ratio, (c) correct sensing ratio, and (d) number of samples.

requires. Meanwhile, decision fusion with a "Majority" rule also showed very good performance in this group of simulations. The three LRTs showed relatively high performance downgrade when always-false SSDF attcks are applied. Although they use different thresholds that governs the tradeoff between false alarm ratios and correct sensing ratios, all of them invariably have much lower correct sensing ratios when N_a increases.

In the set of simulations shown in Fig. 5.3, always-free SSDF attacks aim to create an illusion for a data collector that there is no primary user. Therefore, the consequence will be increased miss detection ratio and decreased false alarm ratio. This has been shown in Fig. 5.3(a) and Fig. 5.3(b). However, one can readily observe that the decision fusion with a "Majority" rule is least robust against always-free SSDF attacks while SPRT, WSPRT, and LRTs are rather stable under the attacks. Remember that miss detection is considered more harmful than false alarm, this can be a major drawback of decision fusion with a "Majority" rule. Because we have chosen to avoid miss detection more than false alarm in SPRT, WSPRT, and LRTs (by assigning lower miss detection probability or higher miss detection cost), these techniques are resilient against always-free SSDF attacks.

5.3.2.3 Impact of Varying Incumbent Signal Strength

In this set of simulations, we fix $N = 500$ and $N_a = 0$ or 100, while varying D from 1000 to 6000 at an interval of 500. When $N_a = 100$, always-false attackers are simulated. Because the value of D decides the expected signal strength a secondary user can receive from the TV tower, the simulations can evaluate the impacts of varying incumbent signal strength on spectrum sensing accuracy. Figs. 5.4(a)-5.4(c) show the results when $N_a = 0$ and Figs. 5.4(d)-5.4(f) show the results when $N_a = 100$. False alarm ratios are not plotted but they can be inferred from miss detection

and correct sensing ratios. From the first three subfigures it can be seen that when there is no attacker, decision fusion with a "Majority" rule has high miss detection ratio in weak received signal scenarios (i.e., D is large), while SPRT, WSPRT, and LRT ($\lambda_2 = 1.6$) achieve better performance in terms of miss detection ratios. This result proves that the mechanisms to limit miss detection in the three fusion schemes are effective even in weak-signal scenarios. The latter three subfigures demonstrate two very interesting phenomena. The first interesting phenomenon is that SPRT's performance (both miss detection ratio and correct sensing ratio) gets better when D increases from 500 to 2,500, which is counter-intuitive. We understand it as the result of insufficient sampling—in Fig. 5.4(f), the number of samples is less than one when $D = 1000$. Because when a secondary user is close to the incumbent transmitter, the likelihood ratio multiplied to S_n in (5.1) tends to be either very large or very small, causing SPRT to accept H_1 or H_0 in very few steps. Therefore, if SSDF attackers happen to contribute to some of the steps, the final sensing result will be distorted. The second phenomenon is that when D is larger than 4,000, all schemes "fail". This is because the CR network becomes out of the incumbent's transmission range. However, when we further look at the curves, it can be found that the schemes "fail" differently. For decision fusion with a "Majority" rule, it in fact outputs fusion results of always "free". Although it fails to detect the primary users, it succeeds in its own objective because in a CR network the objective of DSS is to find if a given location's neighborhood has sensible incumbent signals. For LRTs, SPRT, and WSRPT, their objectives are different—they try to decide if the primary user of interest is in operation. Because when D increases the likelihood ratios for H_0 and H_1 become very close, it is difficult to make reliable fusion decisions. These techniques fail because their objectives are impossible to realize. In fact, when a secondary user has the knowledge of those parameters to calculate *a priori* probabilities in (5.4)-(5.7), it can be calculated whether the secondary user is within the primary user's transmission range. If it is within the transmission range, then LRTs, SPRT, and WSPRT should be used; otherwise, a decision of "free" should be directly made.

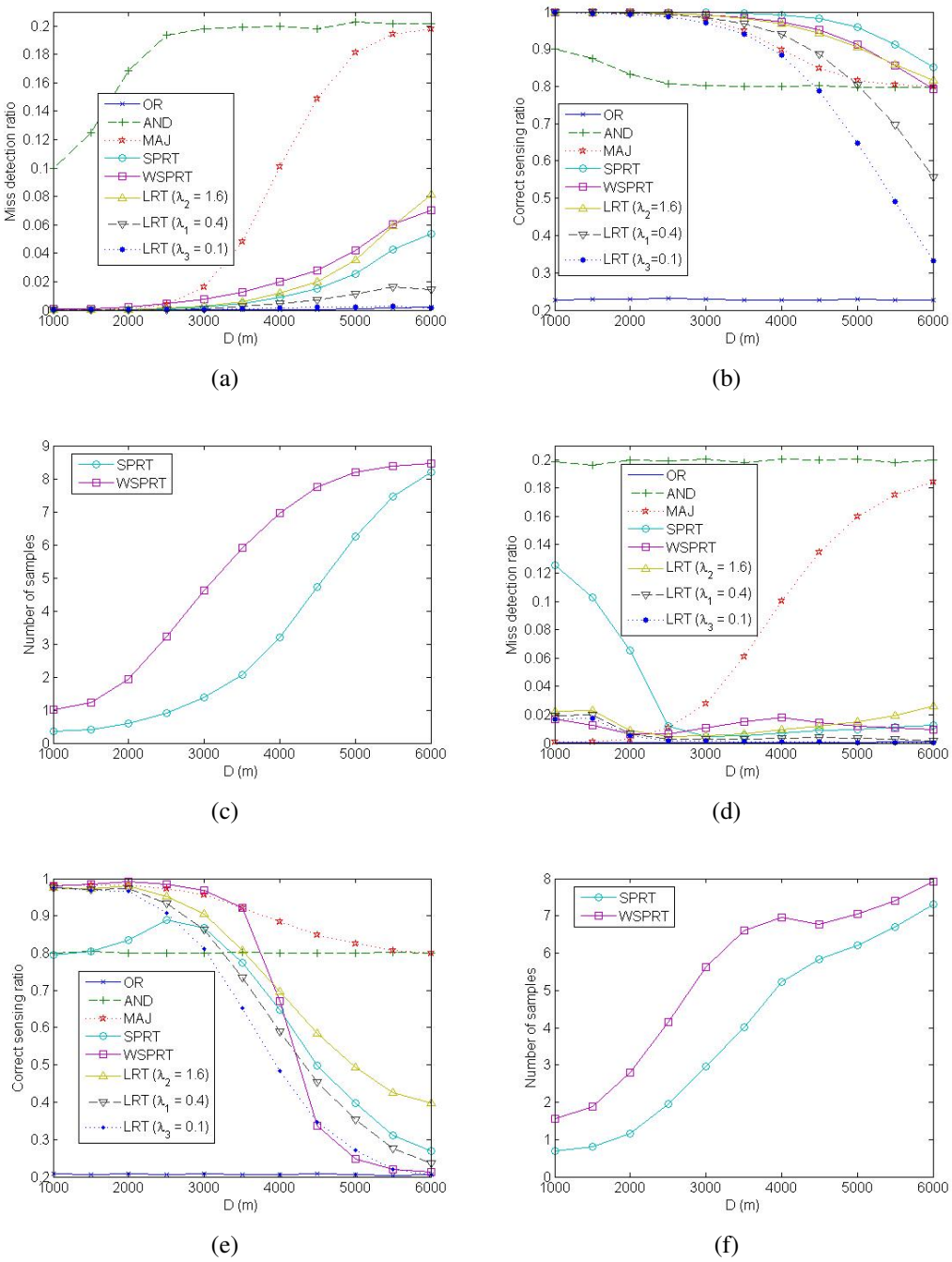


Figure 5.4: The performance of eight fusion techniques with different distances from the simulated network to the TV tower: (a) miss detection ratio when there are no attackers,(b) correct sensing ratio when there are no attackers, (c) number of samples when there are no attackers, (d) miss detection ratio when there are 100 always-false SSDF attackers, (e) correct sensing ratio when there are 100 always-false SSDF attackers, and (f) number of samples when there are 100 always-false SSDF attackers.

5.3.2.4 Impact of Varying Node Density

In this set of simulations, we fix $D = 3000$ and $N_a = 0$ or $0.1N$, while varying N from 50 to 800 at an interval of 50. When $N_a = 0.1N$, always-false attackers are simulated. The value of N in fact decides the node density in an ad hoc CR network. As Fig. 5.5 shows, when N increases, which means the node density increases, the overall correct sensing ratio for all techniques but decision fusion with an "OR" rule or an "AND" rule improves. These two techniques are exceptions since when there are more nodes it is more likely that a data collector receives sensing reports indicating both "1" and "0", which lead to fusion decisions of always "busy" or always "free". SPRT and WSPRT have highest correct sensing ratios but with the cost of increased number of samples. It can be seen from Fig. 5.5(b) and Fig. 5.5(d) that when node density is low, SPRT and WSPRT will automatically collect more samples to compensate. Comparing Fig. 5.5(a) with Fig. 5.5(c), we can see that all curves move downward. However, WSPRT is least affected, obviously benefiting from its weight scheme.

5.4 Practical Considerations

Previous simulation results show that there is no universally optimal fusion technique. One has to find a technique that is suitable for specific applications and meets particular requirements. In this section, we discuss the potential factors that may impact the application of different fusion techniques. In particular, we address several practical concerns including local spectrum sensing techniques, requirement analysis for fusion techniques, security considerations, and 802.22 WRANs.

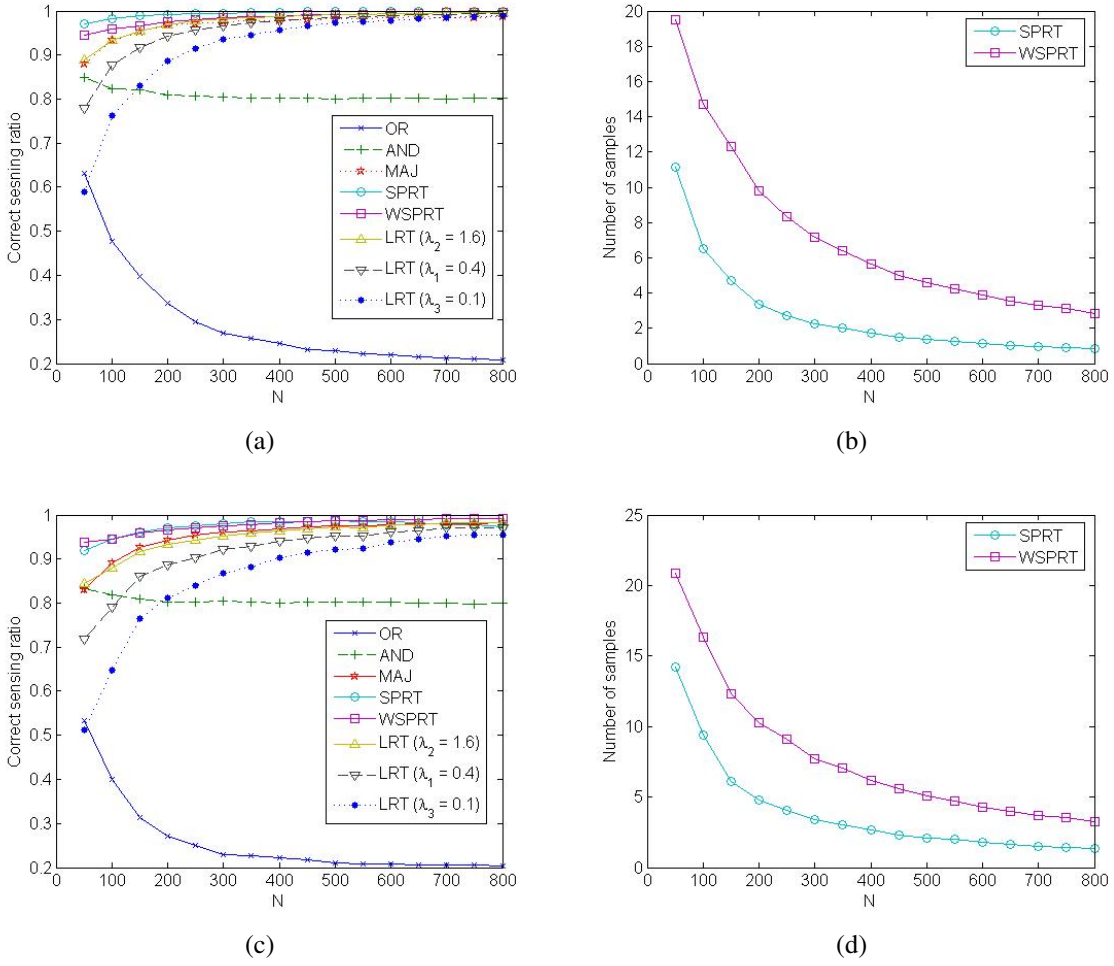


Figure 5.5: The performance of eight fusion techniques when the number of nodes in the network changes: (a) correct sensing ratio when there are no attackers, (b) number of samples when there are no attackers, (c) correct sensing ratio when there are 10% always-false SSDF attackers, and (d) number of samples when there are 10% always-false SSDF attackers.

5.4.1 Impact of the Local Spectrum Sensing Technique

In the previous discussions, we have assumed that an energy detector is used for local spectrum sensing. However, there are alternative spectrum sensing techniques, i.e., matched filter and cyclostationary feature detection [11]. Different from the energy detector, these two techniques are less affected by environmental noise. For example, cyclostationary feature detection is free of the interference caused by Gaussian noise. In this case, false alarm probabilities will be close to zero, i.e., $P(u_i = 1|H_0) = 0$ and $P(u_i = 0|H_0) = 1$. This will cause a problem for all fusion techniques that use a likelihood ratio test (no matter whether there is a fixed or variable number of samples). In particular, if a sensing terminal sends $u_i = 1$, $\frac{P[u_i|H_1]}{P[u_i|H_0]} \rightarrow +\infty$, which leads to a fusion decision of H_1 for all of Bayesian detection, Neyman-Pearson test, SPRT, and WSPRT techniques. On the other hand, when all u_i's are equal to zero, a fusion decision of H_0 is reasonable. As a result, all fusion techniques that use a likelihood ratio test, in spite of higher complexity, generates exactly the same spectrum sensing results as decision fusion with an "OR" rule. Therefore, when sensing terminals use spectrum sensing techniques that are more advanced than simple energy detection, decision fusion techniques are advantageous.

5.4.2 Impact of the Fusion Technique

All fusion techniques other than decision fusion techniques are useful when energy detectors are used in a noisy environment. For example, previous simulation results showed that when there are strict miss detection limitations, Bayesian detection, Neyman-Pearson test, SPRT, and WSPRT are good candidates. And when SSDF attacks are considered, WSPRT is the most favorable. However, these techniques pose additional requirements. These requirements need to be thoroughly investigated before actual usage of a fusion technique. The first requirement is the knowledge of

a priori probabilities. As discussed before, existing research assumes that this knowledge can be obtained from empirical data using on-site calibration. This can be satisfied only when a sensing terminal is static and the empirical data is available. When a sensing terminal is mobile, we have proposed a way to calculate the *a priori* probabilities in 5.1. Although on-site calibration is no longer required, the calculation requires the knowledge of a few environment parameters to be acquired from off-site calibration, i.e., δ, δ_x, P_t, and $\bar{n}_0$ are measured for a given environment (such as rural or urban) and they do not need to be re-calibrated as a sensing terminal moves within the environment. However, we note that this is built upon the assumption that the primary user's location is known. While this requirement can be readily met if TV systems are considered, the more mobile primary users, including Part 74 devices and possibly mobile phones in future, raise an issue. The possibility of multiple primary users in a given spectrum band also complicates the issue. How to deal with these issues are still open problems. The second requirement is the number of samples needed for data fusion, which is specific to techniques using variable number of samples, including SPRT and WSPRT. As shown in 5.3, their better performance in correct sensing ratio comes with the cost of greater number of samples, which means more energy consumption for spectrum sensing, more control overhead for data exchange, and more latency for a decision making process. Therefore, SPRT and WSPRT can be favored only when the additional cost can be tolerated.

5.4.3 Security Considerations

SSDF attacks stem from the fact that the data collector collects spectrum sensing data from sensing terminals and the data may be falsified. For the same reason, when there are other data that sensing terminals need to transmit to the data collector, the possibility of falsifying data should also be considered. As shown in Subsection 5.1, for a data collector to calculate *a priori* probabilities of

a sensing terminal, the knowledge about the sensing terminal's location is required (assume that each secondary user's sensitivity is the same). If the location information is provided by the sensing terminal itself, then a malicious user can set the values of a likelihood ratio and thus arbitrarily manipulate the final sensing decision. To avoid such an attack, one solution is to require a sensing terminal's location to be obtained using certain secure localization schemes, such as those discussed in [10]. Another simpler solution is to use the data collector's own *a priori* probabilities for those of all sensing terminals. This is justified because in an ad hoc CR network two neighboring secondary users are relatively close and they have similar expected received signal strength from a faraway transmitter. And the resulting error from the approximation for each sensing terminal will cancel out with each other when a data collector collects reports from multiple sensing terminals. To verify this idea, we repeated the simulation in Fig. 5.2 and Fig. 5.3 using each data collector's own *a priori* probabilities for its sensing terminals' *a priori* probabilities. For example, Fig. 5.6 plots the correct sensing ratio of the fusion techniques when always-false SSDF attacks are in place. The figure was generated when each data collector replace its sensing terminals' *a priori* probabilities with its own. Comparing Fig. 5.6 and Fig. 5.2(c), we found that the replacement virtually incurs no change to the result.

5.4.4 Considerations for 802.22 WRANs

A final consideration is about 802.22 WRAN networks. As mentioned previously, a WRAN is a cellular network with a cell spanning a circular range that has a radius from 15 to 100 kilometers. A WRAN cell is composed of a BS and numerous CPEs, where the BS is a master that manages the WRAN while the CPEs are slaves that directly communicate with the BS. Typically the BS is the data collector and all CPEs are sensing terminals. A WRAN is different from an ad hoc CR network in that each wireless link in a WRAN spans a much larger distance than a link in an

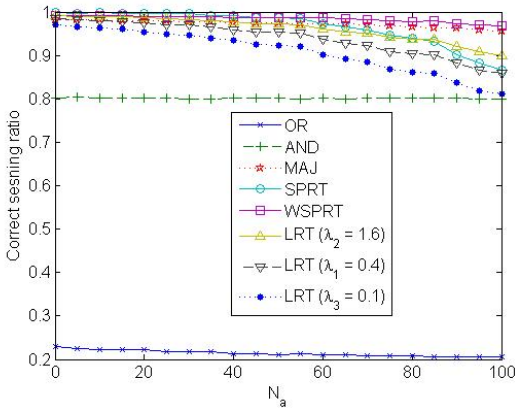


Figure 5.6: The correct sensing ratio of eight fusion techniques as a function of the number of always-false SSDF attackers. Each data collector substitutes its own *a priori* probabilities for its sensing terminals'.

ad hoc CR network does. While a short wireless link is likely to either completely fall within or completely fall out of a primary user's transmission range, a long wireless link is likely to partially fall within the primary user's transmission range. Therefore, it becomes highly possible that even if both a CPE and a BS conduct spectrum sensing correctly, they may produce inconsistent local spectrum sensing results. This innate inconsistency has some implications. Most importantly, even if a CPE's report is different from all others, one cannot simply conclude that it is misbehaving, because it is also possible that the CPE is the only one that falls within a primary user's transmission range. In fact, when calculating reputations for CPEs, a BS should also take each CPE's location into consideration. For example, when a CPE fails to detect a primary user, it is reasonable to decrease its reputation value more if it is close to the incumbent transmitter and keep the reputation value unchanged if it is out of the transmission range of the incumbent transmitter. Again, this problem can be complicated if the primary user is highly mobile or there are multiple primary users. Another implication of a wireless link's large distance is that a BS can no longer use its *a priori* probabilities to approximate a CPE's. As discussed previously, now the solution to securely calculate *a priori* probabilities must apply secure localization schemes.

5.5 Chapter Summary

To overcome the deficiency of existing data fusion techniques discussed in 3.2, this chapter proposed to use SPRT for realizing variable number of sampling and suggest WSPRT that further adds a reputation-based mechanism. With the features of variable number of sampling and the reputation-based mechanism, WSPRT is a data fusion technique robust against SSDF attacks. This chapter also compared all fusion techniques using simulation and analysis. It was shown that decision fusion techniques are easy to implement and can generate correct spectrum sensing decisions that can meet non-strict requirements. Bayesian detection, Neyman-Pearson test, SPRT, and WSPRT are more difficult to realize but can achieve better performance, particularly in terms of miss detection ratios. Among them, WSPRT is the most robust against SSDF attacks, with more system overhead as its cost.

Chapter 6

A Multiple-Rendezvous Cognitive MAC Protocol Robust against CCJ Attacks

In Chapters 2 and 3, we stated that jamming attacks are a security threat to almost all wireless networks. Specifically for CR networks, a CCJ attack could potentially be very disruptive. However, existing cognitive MAC proposals are vulnerable to the attack because they rely on a common/local control channel to exchange control information. This chapter introduces a new cognitive MAC protocol in which the control information exchanged among secondary users is distributed in multiple channels, which we call a MRCMAC protocol. Because in MRCMAC there is no dedicated control channel, it is much more robust against CCJ attacks compared to other cognitive MAC protocols.

The rest of the chapter is organized as follows. We introduce the objectives of MRCMAC in Section 6.1. In Section 6.2 we describe MRCMAC in detail, and in Section 6.3 we analyze MRCMAC and propose several ways to enhance MRCMAC. We evaluate MRCMAC using simulations in Section 6.4 and lastly we summarize the chapter in Section 6.5.

6.1 The Objectives of MRCMAC

Although the proposed MRCMAC is intended to be robust against CCJ attacks, it should not be at the cost of sacrificing network performance or imposing a heavy burden on a CR network. More importantly, the design of MRCMAC needs to consider the special characteristics of a CR network. For these reasons, we set up the following objectives for MRCMAC:

- MRCMAC should consider primary user activities. The most challenging problem caused by primary user activities is spectrum variability. The spectrum variability in CR networks has dual meanings—one is spectrum heterogeneity and the other is spectrum mobility. Spectrum heterogeneity means that spectrum availability in a CR network could change with location, and spectrum mobility means that spectrum availability could change with time. When there is channel variability detected via spectrum sensing, MRCMAC should be able to immediately change the opportunistic channel for transmitting control frames and data frames, avoiding interfering with primary users[1]. This is a fundamental requirement posed by the DSA paradigm. On the other hand, when spectrum variability happens, MRCMAC should not cause a *logical partition*, which stands for a situation when two secondary users are within the transmission range of each other and have at least one common available channel in between, but they are unable to find a channel to communicate with each other.

- MRCMAC should support coexistence of multiple CR networks. In particular, the channel scheduling of MRCMAC needs to ensure that different CR networks minimize interference with each other. To meet this requirement, MRCMAC should have the mechanism to let different CR networks work in orthogonal channels for most of the time.

[1]It is interesting to note that if a jammed channel is considered by secondary users as unavailable channels, the jamming attack becomes a successful PUE attack (see 3.1). In this case, as long as a cognitive MAC protocol is robust to channel variability, it is also robust against a CCJ attack.

- MRCMAC should support channel bonding. In an 802.22 WRAN [16], channel bonding refers to the technique that groups multiple (up to three) contiguous opportunistic channels together and hence increases data rate. MRCMAC should also support channel bonding so that a CR's agility can be fully utilized to maximize network throughput.

- MRCMAC should require only a single radio for each secondary user. Some existing research ideas have proposed that each node be equipped with multiple radios [41, 53]. Multiple radios consume more energy while energy consumption remains a significant constraint in mobile networking scenarios. By using a single radio interface, MRCMAC would face less technical difficulty in deployment compared with schemes that require additional hardware.

- MRCMAC should maximally utilize established technologies. For example, the 802.11 [29] recommended RTS-CTS-DATA-ACK mechanism and related standards are technologies that have been proven to work well in practice. Employing such mechanisms and standards in MRCMAC not only ensures technical correctness, but also allows easy deployment.

In the next section, we detail the design of MRCMAC. We will show how these objectives shape the design of MRCMAC and how MRCMAC combines several novel techniques to attain the design goals.

6.2 MRCMAC

6.2.1 Overview and Assumptions

MRCMAC is a channel hopping scheme that switches each CR across multiple opportunistic channels. It implements three aspects of channel hopping: 1) a pseudo-random channel hopping schedule and the way to synchronize the schedule among all nodes, 2) a method to exchange channel availability information in a CR network, and 3) a channel bonding rule that enables secondary users to access multiple opportunistic channels simultaneously. Unlike existing Cognitive MAC schemes, MRCMAC keeps every node hopping among opportunistic channels even when channel availability does not change. While channel hopping adds some overhead to a CR network, it enables fast response to channel variability. Moreover, the performance exhibited by MRCMAC (see Section 6.4) shows that the overhead is relatively small.

In the following discussion, we assume that there are P contiguous 54Mbps opportunistic channels for a CR network to access. The value 54Mbps is also the maximum data rate in IEEE 802.11a. In Section 6.4, we will use 802.11a as the benchmark protocol to evaluate MRCMAC. The P contiguous channels are numbered $0, 1, 2, \ldots, P-1$, in the order from the lower frequency range to the higher frequency range.

We define a *slot* in MRCMAC to be the time duration each node accesses a channel before hopping to another. During a slot, nodes use the 802.11 recommended Distributed Coordination Function (DCF) protocol based on the RTS-CTS-DATA-ACK mechanism to control access to medium. Previous research [3] shows that for a data rate of 54Mbps, a slot duration of 10ms is long enough to amortize the overhead of channel hopping[2]. Also, we found that the value of 10ms is short enough

[2]Note that when the data rate changes, the slot duration should also be changed so that the expected number of bits to be transmitted in a slot remains constant [3].

for MRCMAC to timely respond channel variability (see Section 6.4.2.3 for details).

MRCMAC is assumed to learn channel availability by spectrum sensing or using other supporting mechanisms (e.g., Radio Environment Map (REM) [79]). Each node uses a P-bit channel vector (CV) to represent the availability of channels. The p-th ($p = 1, 2, \ldots, P$) least significant bit in the CV represents whether the channel numbered $(p-1)$ is available or not—a set bit indicates that the corresponding channel is available, and a cleared bit indicates that the channel is unavailable. We assume that the CR network performs DSS (see 2.2.2). In DSS, each node not only considers its own channel availability, but also considers its neighboring nodes'. Therefore, MRCMAC requires each node to broadcast its CV to neighboring nodes and store a local copy of CVs that are received from neighboring nodes. A channel is considered available only when its corresponding bits in both its own CV and all received CVs are set (i.e., the decision fusion with an "OR" rule is used).

Like in the 802.22 standard, in MRCMAC we assume that each radio can bond up to three channels, and the bonded channels must be contiguous. Each MRCMAC node is equipped with a single half-duplex radio.

6.2.2 The Basic Pseudo-random Channel Hopping Scheme

For the sake of discussion, we make several assumptions in this subsection. First, all nodes in a CR network are assumed to be synchronized. However, in Section 6.3, we will show how MRCMAC can be independent of the assumption by enabling all nodes to gradually synchronize to each other in a dynamic fashion. Second, we assume that all CV bits are set. This assumption will no longer be needed in Subsection 6.2.3 when we consider channel variability. Finally, P is assumed to be a prime number. Again, in Section 6.3, we will generalize the number of opportunistic channels to an arbitrary value.

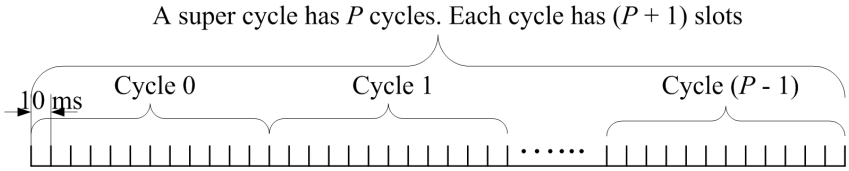


Figure 6.1: The concept of a cycle and a supercycle.

In the basic pseudo-random channel hopping scheme, node N_i ($i = 1, 2, \ldots, N$, where N is the number of nodes in a CR network) randomly chooses two numbers: a starting integer channel index s_i that is randomly chosen from $[0, P-1]$, and an integer hopping offset o_i that is randomly chosen from $[1, P-1]$. We define a *cycle* to be $P+1$ slots, as Fig. 6.1 shows. For the j-th slot ($j = 0, 1, \ldots, P$) in a cycle, node i chooses the channel index as:

$$c_{i,j} = \begin{cases} (s_i + j \cdot o_i) \bmod P & (j < P) \\ o_i & (j = P) \end{cases} . \tag{6.1}$$

The above channel schedule guarantees that if two nodes' time is synchronized, they will hop to the same channel at least for one slot in a cycle. Suppose we have two nodes denoted as N_1 and N_2, respectively. The above statement is obviously true when $o_1 = o_2$ since at the P-th slot the two nodes will switch to the same channel. If $o_1 \neq o_2$, without loss of generality, we define $\Delta o = o_1 - o_2 > 0$ and $\Delta s = s_1 - s_2$. The modular difference between $c_{1,j}$ and $c_{2,j}$ in the first P slots form the set $Z_P = \{(\Delta s + j \cdot \Delta o) \bmod P : j = 0, 1, \ldots, P-1\}$. It is known that Z_P is an order-P group under the binary operation "additive modulo P", and the group has an identity member being zero [68]. This means that the modular difference between $c_{1,j}$ and $c_{2,j}$ will be zero for exactly one slot in the first P slots. In that slot nodes N_1 and N_2 will switch to the same channel. We use an example to show that this is true. Assume that $P = 5$, $s_1 = 1$, $s_2 = 2$, $o_1 = 2$, and $o_2 = 3$. According to (6.1), the channel schedules of N_1 and N_2 in a 6-slot cycle will be $(1, 3, 0, 2, 4, 2)$ and $(2, 0, 3, 1, 4, 3)$, respectively. Obviously, they will encounter each other at the fifth slot in channel four.

In every slot, each node will broadcast a Neighbor Discovery (ND) frame in the selected channel. The ND frame includes the information of the sending node's s_i, o_i, and a locally measured P-bit CV v_i. We assume that all nodes in a CR network share a function $f(s_i, o_i)$ that defines an order among all s_i and o_i pairs. The function should output different values for any different s_i and o_i pairs. For example, $f(s_i, o_i) = s_i \cdot P + o_i$ is a function satisfying the requirement. When a node N_1 receives an ND frame from node N_2, N_1 checks if the received s_2 and o_2 are the same as s_1 and o_1. If they are different and $f(s_2, o_2) > f(s_1, o_1)$, N_1 updates its s_1 and o_1 using s_2 and o_2. Otherwise, the values of s_1 and o_1 remain unchanged. This mechanism for updating channel index and hopping offset ensures that all nodes will eventually share the same s_i and o_i, which is the pair that maximizes the $f(s_i, o_i)$ value in the network. For different CR networks, the function $f(s_i, o_i)$ should be chosen differently so that two networks are not going to follow the same channel hop sequence. Making the function different leads to frequency diversity and this will be helpful to coexistence of multiple CR networks.

When nodes share the same s_i and o_i, they enter a *synchronous hopping state* in which all nodes access the same channel in every slot. In every slot of the synchronous hopping state, all nodes simply use the RTS/CTS/DATA/ACK mechanism recommended by 802.11 to communicate with each other. However, a special care needs to be taken at a slot boundary. If a node is transmitting or receiving a frame when a slot ends and the channel is about to switch, MRCMAC stops transmitting or receiving. The unsent network-layer packet will be sent again in the next slot.

6.2.3 CV exchange

The previous discussion did not consider channel variability. In CR networks, however, the availability of opportunistic channels is expected to often change. The v_i in every broadcast ND frame contains the CV that is measured locally by the sending node. Every node stores a local copy of

all the CVs it has received. The local copy will be updated if a newer CV is received from the same sending node. When a node needs to check its channel availability, it calculates an aggregate CV as a logical OR over its own measured CV and all the stored CVs. Only the channels whose corresponding bits in the aggregate CV are set are regarded as available. As discussed in Subsection 6.2.2, when all channels are available, a node N_i uses (6.1) to choose the channel index to switch into. However, if the selected channel is not available in the final CV, MRCMAC requires N_i to repeat the computation

$$c_{i,j} \leftarrow (c_{i,j} + R) \bmod P \tag{6.2}$$

until $c_{i,j}$'s corresponding bit (i.e., the $(c_{i,j} + 1)$-th least significant bit) in the aggregate CV is set. Here R is an integer constant randomly chosen from $[1, P - 1]$ and is known to all nodes in a CR network. The property of additive modular operation dictates that this method defines a search sequence that traverse all P channels, and a different R defines different search sequences. If no channel is found to be available after searching all channels, node N_i simply refrains from transmitting. The use of the random number R, similar to $f(s_i, o_i)$, is helpful to the coexistence of different CR networks—when two CR networks happen to share the same s_i and o_i, if they choose different values of R and thus define different channel search sequences, there is better chance for them to access different channels, i.e., frequency diversity is enhanced.

If we consider spectrum heterogeneity, the above neighbor discovery process could be problematic. Suppose based on the channel hopping scheme defined by (6.1), two nodes N_1 and N_2 should encounter each other at channel C_x during a certain slot. Because of spectrum heterogeneity, C_x may be available to N_1 but not available to N_2. Then N_1 will simply access C_x while N_2 will search an available channel to access using (6.2), which will be different from C_x. In the basic channel hopping scheme, it is possible that N_1 and N_2 can only meet each other at channel C_x while they actually share another commonly available channel C_y. In that case, the problem of

logical partition would arise. Therefore, we need to improve the basic channel hopping scheme so that every time N_1 and N_2 switch to a common channel, the channel will be different. For this purpose, we define a supercycle as P contiguous cycles. As Fig. 6.1 shows, each cycle can be indexed k ($k = 0, 1, \ldots, P-1$) in the supercycle it belongs to. Now we update (6.1) as

$$c_{i,j,k} = \begin{cases} (k + s_i + j \cdot o_i) \bmod P & (j < P) \\ (k + o_i) \bmod P & (j = P) \end{cases} . \tag{6.3}$$

Obviously, because a shift value k is added, N_1 and N_2 will encounter each other at a different channel at a cycle with a different k value.

In addition to exchanging channel vectors among neighboring nodes, it is necessary for each node to forward its neighboring nodes' CVs to farther nodes. This requirement arises from the need to minimize interference to primary users. In a CR network, when a node detects a primary user's presence using spectrum sensing, it detects the signal of the primary transmitter but not the primary receiver. To protect the primary receiver from being interfered by secondary users, there should be a margin area around where the primary transmitter is detected and no secondary user should be allowed for any co-channel operation within the margin area. For example, in the 802.22 WRAN standard, a 4.7km distance is proposed as the radius of the margin area for analog TV primary users [16]. Therefore, in MRCMAC, when an opportunistic channel is detected to be busy at a node, it is necessary for not only single-hop, but multi-hop neighboring nodes not to access the channel.

Although each node needs to forward other nodes' CVs, it is prohibitive to forward all of them in every time slot. To solve this problem, we propose a technique called "selective CV forwarding". In this technique, besides the CV of the sending node itself, each ND frame will piggyback another (and at most one) node's CV information, which includes the node's ID/MAC address (denoted as

N_f), the node's CV v_f, and the time duration the CV has been received t_f (in number of slots). To choose which stored CV to forward, a node divides all stored CVs into two groups: one group is called "fresh CVs", each of which has been forwarded less than three times. The remaining CVs fall into the "old CVs" group. The number "three" was chosen since it proved to be effective in our simulation experiment (see Section 6.4 for more details). If the "fresh CVs" group is non-empty, the node randomly chooses one CV out of the group to forward. Otherwise, the node randomly chooses one CV out of the "old CVs" group to forward if the "old CVs" group is non-empty. If the "old CV" is also empty, an ND frame piggybacks nothing. Using "selective CV forwarding", MRCMAC is able to propagate most recently updated CVs the fastest while keeping the forwarding overhead relatively low. In Section 6.4, the simulation results show that MRCMAC is scalable when the number of nodes in a CR network increases. This is partially attributed to the "selective CV forwarding" technique.

6.2.4 Support for Channel Bonding

MRCMAC uses a simple greedy method for channel bonding. Once $c_{i,j,k}$ is derived from (6.3), a node checks if its two adjacent channels with indexes $c_{i,j,k} - 1$ and $c_{i,j,k} + 1$ are available (if $c_{i,j,k} - 1 < 0$ or $c_{i,j,k} + 1 > P - 1$, the corresponding channel is considered not available). When either adjacent channel is available, it is added to the bonded channel. Therefore, depending on the availability of the adjacent channels, the number of bonded channels could be three, two, or only one.

Such a design is valid if the nodes in a CR network already successfully exchanged CVs and share the same information regarding channel availability. This condition is most likely to be true for a CR network that has abundant opportunistic channels to access, because in that case two neighboring nodes have high probability to find at least one channel that results in the same bonded

channels. However, in certain extreme cases, the simple greedy method for channel bonding could be problematic. For example, suppose there are two neighboring nodes N_1 and N_2. N_1 only has Channel 0 available while N_2 has Channel 0, 1, and 2 available. Using the simple greedy method, before the nodes discover each other, N_1 always operates in Channel 0, but at any given time, N_2 will access at least two bonded channels given the simple greedy method. As a result, N_1 and N_2 will never be able to encounter each other, causing a logical partition. To solve the problem, we introduce a mechanism called "periodic channel bonding disabling" (PCBD). The idea is that for every $P + 1$ cycles, during a whole cycle all nodes only choose a single channel to access but do not use channel bonding. This means that if PCBD is conducted during a cycle with an index of k within a supercycle, the next cycle to conduct PCBD has an index of $((k + 1) \bmod P)$ within a supercycle. Therefore, every time PCBD is conducted, two nodes that have not synchronized their hopping sequence will encounter each other at a different channel. As long as the two nodes share a common available channel, they will successfully discover each other in a slot during a PCBD cycle. When the value of P is sufficiently large, the throughput loss due to PCBD is negligible (See Section 6.4 for more details). When the value of P is small, one can also adjust the period of PCBD from $P + 1$ to $2P + 1$ or $3P + 1$ so that the throughput loss caused by PCBD can be decreased.

6.3 Enhancement and Analysis

In Section 6.2 we described the MRCMAC under some assumptions. However, some of the assumptions can be relaxed. In this section, we discuss the ways to avoid the requirement that all nodes be synchronized offline and to generalize P to an arbitrary number. In addition, we consider some practical issues that will be important for MRCMAC implementation.

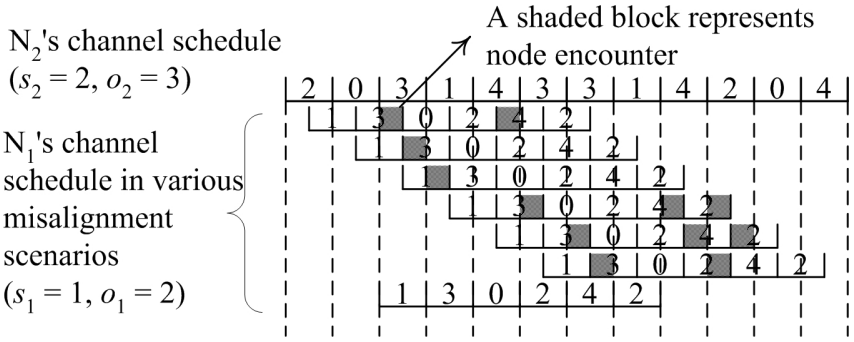


Figure 6.2: When two nodes are not synchronized, there is still a good chance for them to encounter in a cycle.

6.3.1 Online Synchronization

Based on (6.3), if the time at two nodes is synchronized, they are guaranteed to share the same $c_{i,j,k}$ at a certain slot within a cycle. However, the requirement of getting all nodes synchronized offline might be expensive and unnecessary. Fig. 6.2 shows an example when $P = 5$ under various misalignment cases. The figure draws two cycles of node N_2 and one cycle of node N_1 which could start at any slot of N_2's cycle. The values of s_1, s_2, o_1, and o_2 are randomly chosen[3]. As the figure shows, even when the two nodes randomly select the time to start channel hopping, in most cases they are able to encounter each other for at least half a slot during each cycle (assuming that the two nodes have the same aggregate CVs). Therefore, the two nodes can synchronize their time by exchanging ND frames during the encounter time, e.g., by embedding time information in ND frames and using a simple averaging scheme such as that proposed in [19]. In this way, MRCMAC could support online synchronization.

However, in the misalignment case, an encounter is not guaranteed. The last scenario in Fig. 6.2 shows an example where N_1 and N_2 will never encounter each other. However, our analytical result shows that the probability of this scenario is very low. In fact, as proved in Theorem I, when

[3]Note that the misalignment of cycle index k is implicitly considered in the analysis. Because in (6.3) any change in k's value could be compensated by altering both s_i's and o_i's values, it is sufficient to only consider the distributions of s_i and o_i.

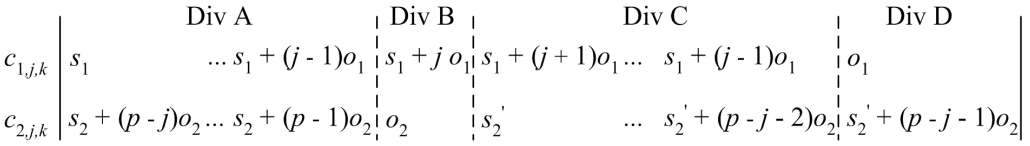


Figure 6.3: The channel schedules of N_1 and N_2 in a complete N_1's cycle ($s_2' = (s_2 + 1) \bmod P$).

two nodes are not synchronized, in each cycle the probability that the two nodes will choose to switch to the same channel (according to (6.3)) for at least half of the time duration of a slot is:

$$P_C \geq 1 - \frac{(P-1)^3}{(P+1)P^3} - \frac{(P-1)\,(P-2)\left(\prod_{i=1}^{(P-1)/2}(P-i)\right)^2}{(P+1)\,P^P} \tag{6.4}$$

Theorem I: Assume nodes N_1 and N_2 are not synchronized and randomly choose their start channel index as s_1 and s_2, channel hopping offset as o_1 and o_2, respectively. Also, assume that the nodes have the same aggregate CV. Then in a cycle, the probability for N_1 and N_2 to encounter each other at a common channel for at least half of the time duration of a slot satisfies (6.4).

Proof: When N_1 and N_2 are not synchronized, the slot boundary of N_1 could split N_2's slot into two parts (as Fig. 6.2 shows). We focus on the part with longer time duration, which has at least half of the time duration of a slot. By ignoring the shorter part, we now can regard the two nodes' slot boundary (but not necessarily cycle boundary) as aligned. We distinguish two scenarios: when the cycles of N_1 and N_2 happen to align to each other and when they are not. The first scenario takes place at a probability of $1/(P+1)$, since one node's cycle could start at any of the $P+1$ slots in the other node's cycle and the alignment happens only if it starts at the first slot. In this scenario, the discussion in Subsection 6.2.2 has shown that N_1 and N_2 must be able to encounter each other within a cycle. The scenario when the cycles of N_1 and N_2 misalign with each other happens at a probability of $P/(P+1)$. Fig. 6.3 shows the two nodes' channel schedule in a complete cycle of N_1. As the figure shows, the cycle overlaps with N_2's two adjacent cycles and it is conceptually cut

into four divisions. Whether there is an encounter in the slots of one division is independent of that in the slots of another division. We further consider two cases—when $o_1 = o_2$ and when $o_1 \neq o_2$. The case $o_1 = o_2$ happens at a probability of $1/(P-1)$, since both o_1 and o_2 are randomly selected from the same set of $P-1$ numbers. In this case, for each of the four divisions there is $(1/P)$ probability that there will be an encounter in the division. (Note that this applies to Div A and Div C because in all slots of Div A or Div C, $c_{1,j,k}$ and $c_{2,j,k}$ keep constant modular-P difference.) Therefore, the probability that there is at least one encounter in all four divisions is:

$$P_{case-1} = 1 - \left(1 - \frac{1}{P}\right)^4. \tag{6.5}$$

When $o_1 \neq o_2$, for Div B and Div D, an encounter still happens at a probability of $1/P$. For Div A and Div C, the probabilities that no encounter happens are $\left(1-\frac{1}{P}\right)\cdot\left(1-\frac{2}{P}\right)\cdot ...\cdot\left(1-\frac{j}{P}\right)$ and $\left(1-\frac{1}{P}\right)\cdot\left(1-\frac{2}{P}\right)\cdot ...\cdot\left(1-\frac{P-j-1}{P}\right)$, respectively. Therefore, in the second case, the probability that there is at least one encounter in all four divisions is

$$\begin{aligned} P_{case-2} &= 1 - \left(1-\tfrac{1}{P}\right)^2 \prod_{i=1}^{j}\left(1-\tfrac{i}{P}\right)\cdot\prod_{i=1}^{P-j-1}\left(1-\tfrac{i}{P}\right) \\ &\geq 1 - \left(1-\tfrac{1}{P}\right)^2 \left[\prod_{i=1}^{(P-1)/2}\left(1-\tfrac{i}{P}\right)\right]^2 \end{aligned} \tag{6.6}$$

Summarizing all scenarios and cases, we have:

$$P_C = \frac{1}{P+1} + \frac{P}{P+1}\left(\frac{1}{P-1}\cdot P_{case-1} + \frac{P-2}{P-1}\cdot P_{case-2}\right) \tag{6.7}$$

Replacing P_{case-1} and P_{case-2} in the above equation with (6.5) and (6.6), we can derive (6.4). QED.

The right hand side of (6.4) is plotted in Fig. 6.4. It shows that at a very high probability any two nodes would encounter in a cycle. Therefore, when a new node joins in a CR network whose nodes are already synchronized, it is very likely for the node to synchronize with all other nodes very soon. If the node is unable to detect other nodes, it simply chooses another pair of s_i and o_i

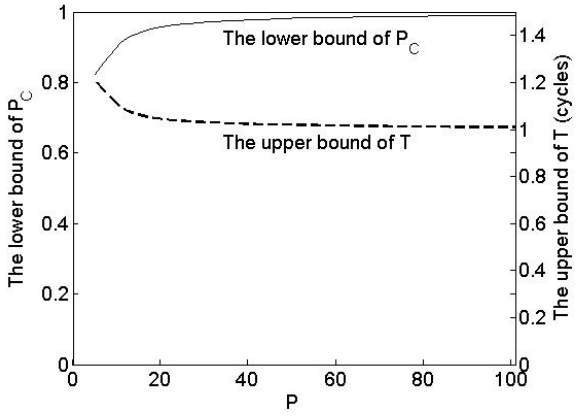


Figure 6.4: Plot the lower bound of P_C and the upper bound of T.

and tries again. This process is repeated until the node gets synchronized with other nodes. The expected time (in cycles) to complete this process is:

$$T \leq \sum_{i=0}^{+\infty} (1 - P_c)^i P_c \cdot i = \frac{1}{P_c} \tag{6.8}$$

which is relatively small as shown in Fig. 6.4. Note that the above analysis has ignored channel heterogeneity. When channel heterogeneity is considered, there will be additional delay in neighbor discovery, whose value will be dependent on the specific channel availability conditions in a CR network.

6.3.2 Arbitrary Number of Channels

In Subsection 6.2.2, we showed that MRCMAC's channel hopping scheme guarantees two nodes' encounter in a cycle if the number of opportunistic channels is a prime number. If one desires to apply MRCMAC to a CR network with a composite number of opportunistic channels (denoted as C), a straightforward way is to choose a prime number greater than C and map the resulting channel index down to the actual number of channels, e.g., using a modular operation. Obviously,

this would cause bias over some channels and require additional bits to represent channel indexes. A more elegant method exists if we factor C into M primes and implement a cycle as M nested subcycles, with each subcycle consuming a prime. We use an example of $M = 3$ to illustrate how this idea works. Let $C = P_1P_2P_3$, where $P_l(l = 1, 2, 3)$ is a prime. Now a cycle becomes $(P_1 + 1)(P_2 + 1)(P_3 + 1)$ slots. Every node selects a starting channel index and a hopping offset for each subcycle. We represent them as three-element tuples (s_i^1, s_i^2, s_i^3) and (o_i^1, o_i^2, o_i^3), where s_i^l is chosen from $[0, P_l - 1]$ and o_i^l is chosen from $[1, P_l - 1]$. Accordingly we update (6.3) as:

$$\begin{aligned} c_{i,j,k} = \quad & \left[\left(c_{i,j}^1 \cdot P_2 + c_{i,j}^2\right) \cdot P_3 + c_{i,j}^3 + k\right] \bmod (P_1 + 1)(P_2 + 1)(P_3 + 1) \\ & (0 \le j \le (P_1 + 1)(P_2 + 1)(P_3 + 1) - 1) \end{aligned} \tag{6.9}$$

where

$$c_{i,j}^l = \begin{cases} (s_i^l + j^l \cdot o_i^l) \bmod P_l & (j^l < P_l) \\ o_i^l & (j^l = P_l) \end{cases} \tag{6.10}$$

and

$$j^l = \left\lfloor \frac{j \bmod \prod_{m=l}^{3} (P_l + 1)}{\prod_{m=l+1}^{3} (P_l + 1)} \right\rfloor. \tag{6.11}$$

In (6.11) the notation $\prod$ is assumed to be one if its upper limit is smaller than its lower limit. Based on the result of Subsection III.1, it is straightforward to recursively prove that during the time when two nodes share a common $c_{i,j}^l$, they must share a common $c_{i,j}^{l+1}$ for some slots. Therefore, $c_{i,j,k}$ ensures an encounter within a cycle. To illustrate this in a more concrete fasion, Table 6.1 shows a numerical example when $C = 15$. In the example, we have $M = 2$, $P_1 = 3$, and $P_2 = 5$. Now a cycle contains $(P_1 + 1)(P_2 + 1) = 24$ slots. Assume that nodes N_1 and N_2 choose their starting channel indexes and hopping offsets as $(s_1^1, s_1^2) = (0, 2)$, $(o_1^1, o_1^2) = (1, 3)$, $(s_2^1, s_2^2) = (1, 4)$, and $(o_2^1, o_2^2) = (2, 4)$, respectively. Without loss of generality, we assume the cycle index k to be zero. The table lists the channel indexes of both nodes during a cycle. As the table shows, a subcycle in the cycle contains $P_2+1 = 6$ slots. The two nodes share the same $c_{i,j}^1$ during the subcycle spanning

Table 6.1: A numerical example of the enhanced channel hopping scheme when the total number of opportunisitc channels is a composite number.

j	0	1	2	3	4	5	6	7	8	9	10	11
$c^1_{1,j}$	0	0	0	0	0	0	1	1	1	1	1	1
$c^2_{1,j}$	2	0	3	1	4	3	2	0	3	1	4	3
$c_{1,j,0}$	2	0	3	1	4	3	7	5	8	6	9	8
$c^1_{2,j}$	1	1	1	1	1	1	0	0	0	0	0	0
$c^2_{2,j}$	4	3	2	1	0	4	4	3	2	1	0	4
$c_{2,j,0}$	9	8	7	6	5	9	4	3	2	1	0	4
j	12	13	14	**15**	16	17	18	19	20	21	22	23
$c^1_{1,j}$	2	2	2	**2**	2	2	1	1	1	1	1	1
$c^2_{1,j}$	2	0	3	**1**	4	3	2	0	3	1	4	3
$c_{1,j,0}$	12	10	13	**11**	14	13	7	5	8	6	9	8
$c^1_{2,j}$	2	2	2	**2**	2	2	2	2	2	2	2	2
$c^2_{2,j}$	4	3	2	**1**	0	4	4	3	2	1	0	4
$c_{2,j,0}$	14	13	12	**11**	10	14	14	13	12	11	10	14

from slot 12 to 17. Within the subcycle, the two nodes share the same $c^2_{i,j}$ at slot 15, which is also the slot when the two nodes will encounter each other. This example shows that MRCMAC can be generalized to the application of any number of opportunistic channels.

6.3.3 Implementation Considerations

Previous discussion has shown that besides the RTS, CTS, DATA, and ACK frames used in 802.11, MRCMAC requires a new type of ND frame to broadcast a sending node's channel schedule information (s_i, o_i, and v_i) and to forward another node's CV information (N_f, v_f, and t_f). In practice, broadcasting a full ND frame might be costly and unnecessary as the information an ND frame contains can be embedded in other frames. According to [3], IEEE 802.11's Long Control Frame Header can be used to embed the information. Different from the commonly used short control frames used in 802.11(e.g., RTS, CTS, and ACK), the long frame header has six unused bytes.

The variables s_i, o_i, v_i, N_f, v_f, and t_f take $\log_2 P$, $\log_2(P-1)$, P, 48, P, and 7 bits[4] to store, respectively. Depending on the value of P, several control frames may be needed to piggyback all the information. One can define the six bytes appropriately so that the cost of sending an ND frame will be amortized to several long control frames. Only when a node does not have enough control frames to send in a slot, would the node broadcast an ND frame. Such an implementation could potentially save a lot of overhead in a dense CR network, making MRCMAC scalable to the number of nodes in the network.

6.3.4 Security Considerations

Because MRCMAC does not require a control channel, it is inherently robust against CCJ attacks. However, an attacker can attempt to enter the synchronous hopping state together with other nodes in a network and launch a jamming attack in every slot. Such an attack can be mitigated by adding a layer of obscurity to the pseudo-random hopping sequences. A simple method is to share a secret key X among all nodes in a CR network. Whenever a node uses (6.3) to calculate a channel index $c_{i,j,k}$, it updates the index as $c_{i,j,k} \leftarrow (c_{i,j,k} + h(X,t)) \bmod P$, where $h(X,t)$ is a one-way hash function that takes X and current time t and outputs an integer from $[0, P]$. The method could be problematic once the secret key is revealed to attackers. It could be strengthened by changing X into a hash chain $X(t)$, which is a chain of pre-calculated hash values that satisfies: $X(t) = h_2[X(t+1)]$. Here $h_2(X)$ is another one-way hash function. By releasing the hash chain in a reverse order, revelation of a key at an early time will not disclose the key to be used later.

[4]In MRCMAC implementation, because DSS should avoid using stale spectrum sensing results, a stored CV is purged if no update has been received for the CV for 100 slots (i.e., t_f takes seven bits). Choosing this number yields good performance of MRCMAC (see Section 6.4).

6.4 Simulation Study

6.4.1 Simulation Setup

We simulate MRCMAC in ns-2 [51] and use the single-channel IEEE 802.11a as the reference protocol for comparison. Note that MRCMAC and 802.11a are fundamentally different in that only MRCMAC can be used in CR networks with multiple opportunistic channels. However, 802.11a serves as a good benchmark to evaluate how well MRCMAC performs. This is because when MRCMAC enters the synchronous hopping state, its major difference from 802.11a lies in MRCMAC's channel switching delay that occurs every 10ms. Using 802.11a as a reference, we are able to tell what overhead the channel switching delay causes and whether it is acceptable.

In the simulation, MRCMAC has P=17 54Mbps opportunistic channels to access. To maximally ensure fair comparison, the opportunistic channels are assumed to reside in the same 5GHz band that 802.11a is using, and 802.11a works at the 54Mbps data rate. MRCMAC allows bonding up to three opportunistic channels. The data rate of bonded channels is proportional to the number of channels. However, the interframe spaces defined in the DCF protocol are kept the same as that in 802.11a[5]. In MRCMAC, the channel switching delay is chosen as $80\mu s$, which is well supported by existing technology [26].

In the simulation, all nodes in a single simulation run either use MRCMAC or 802.11a as the MAC protocol. Every node uses Dynamic Source Routing (DSR) [33] as the routing protocol. At the transport layer, the User Datagram Protocol (UDP) is used in the simulations by default. However, MRCMAC's performance under the Transmission Control Protocol (TCP) is also evaluated (see Subsection 6.4.2.2). The traffic generator uses Constant Bit Rate (CBR) flows of 1,500-byte

[5]Different from data rate, the interframe spaces are mainly decided by a radio's hardware limitation, i.e., the minimal time to switch between its transmitting mode and its receiving mode.

segments sent at a 200μs interval. This traffic load is more than what can be supported in every simulated network.

The simulation results to be presented in 6.4.2 can be divided into two types. The first type of results are shown in figures with the simulation time being the x-axis. These figures show the result of a single simulation run. They provide a micro-level view of throughput variations in MRCMAC or 802.11a. The rest of figures show the statistical results that have been obtained from the simulation. In these figures, every datum is the average of the results derived from ten independent simulation runs, with each simulation run being 900s. We have found that this simulation time yields stable simulation results. For example, in Fig. 6.6(a) and Fig. 6.6(b) the error bar we plotted nearly overlaps with the average value.

In our simulation, we also used the random waypoint mobility model (see 3.1). Similar to the simulations conducted in 3.1, we simulated some time before the real simulation to ensure that the random waypoint model entered steady state. The parameter used in the mobility model will be elaborated in the simulations of 6.4.2 that used the model.

6.4.2 Simulation Results

In this subsection, we present the simulation results of MRCMAC. In particular, we study MRCMAC's neighbor discovery delay, multi-hop network throughput under UDP and TCP, and MRCMAC's performance under varying conditions, including spectrum variability, node mobility, clock drift, and multi-flow complex networks.

6.4.2.1 Neighbor Discovery Delay and Single-hop UDP Throughput

In MRCMAC, each node randomly chooses a starting channel index and a hopping offset. It takes some time for any two nodes to encounter each other and synchronize their channel schedules. The time taken for all nodes in a CR network to synchronize their channel schedules is called neighbor discovery delay. Figs. 6.5(a) and 6.5(b) show the microscopic view of UDP throughput in a single-hop network when the PCBD mechanism is enabled or disabled, respectively. In both cases, we see that the delay for MRCMAC to generate any throughput is approximately 0.40s. Although this value is relatively small, it is greater than what we expected. According to the analysis in Subsection 6.2.2, two neighboring nodes should encounter each other and complete neighbor discovery within one cycle, i.e., the worst case should be 0.18s. Further investigation shows that the two nodes actually entered the synchronous channel hopping state much earlier, but in our simulation the Route Request packet in DSR has a retransmission delay of 0.40s[6]. If we consider the real neighbor discovery delay, Figs. 6.5(c) and 6.5(d) show the results of a multi-hop network case and a random network case. In the latter case, a varying number of nodes are randomly placed in a 2000m$\times$2000m network area[7]. The results of Figs. 6.5(c) and 6.5(d) were collected from ten independent simulation runs. One can observe that the worst-case single-hop neighbor discovery delay is indeed 0.18s and the average neighbor discovery delay for any case is typically no more than one second. An interesting phenomenon exists in Fig. 6.5(d) where the neighbor discovery delay first increases, then decreases as the number of nodes increases. It results from the fact that the neighbor discovery delay directly relates to the greatest number of hops in a CR network and that the greatest number of hops first increases, then decreases as the number of nodes increases.

[6]This delay is not defined in the DSR standard but is implementation-specific. In practice, the value could be shortened to enable faster route discovery.

[7]The random placement may result in physical partition of the CR network. In that case, we calculate the neighbor discovery delay as the time taken for all nodes in every partition to start synchronous channel hopping.

Figs. 6.5(a) and 6.5(b) also show the throughput MRCMAC can achieve. MRCMAC's throughput gain over the single-channel 802.11a is the result of channel bonding. However, since we allow up to three channels to be bonded, the throughput gain is not proportional to the number of bonded channels. Recall that the interframe spaces used in MRCMAC has been the same as those used in 802.11a. The constant overhead the spaces impose is the major reason that the throughput does not scale with the number of bonded channels. A minor reason lies in the channel switch delay (80μs for every 10ms-slot). The delay causes the last packet sent in a slot to be sent again in the next slot. We find that in our simulation a node can in each slot send approximately 16 packets when a single channel is used and send 20 packets when three bonded channels are used, that means a roughly 5%-6% throughput loss due to the channel switch delay. Comparing the throughput at the ditches (caused by PCBD) in Fig. 6.5(b) and that of 802.11a, we can observe a similar amount of throughput loss.

If we only look at 802.11a's throughput, we found that the results have been consistent with other experimental results that have been presented in the literature (e.g., [3]), even though a different simulation tool may have been used (the simulation in [3] used QualNet). This also indirectly shows the correctness of our simulation.

6.4.2.2 Multi-hop Network Throughput

Previous simulations have shown the UDP throughput in a single-hop network. Our next experiment is to study both the UDP and the TCP[8] throughput in a single-flow multi-hop network. Fig. 6.6 shows the simulation results, which were averaged over ten independent simulation runs. In Fig. 6.6(a), MRCMAC's overall UDP throughput is not much affected when the PCBD mechanism is enabled. Therefore, in all the remaining simulations using UDP, we always enable

[8]We used TCP Newreno in our simuation.

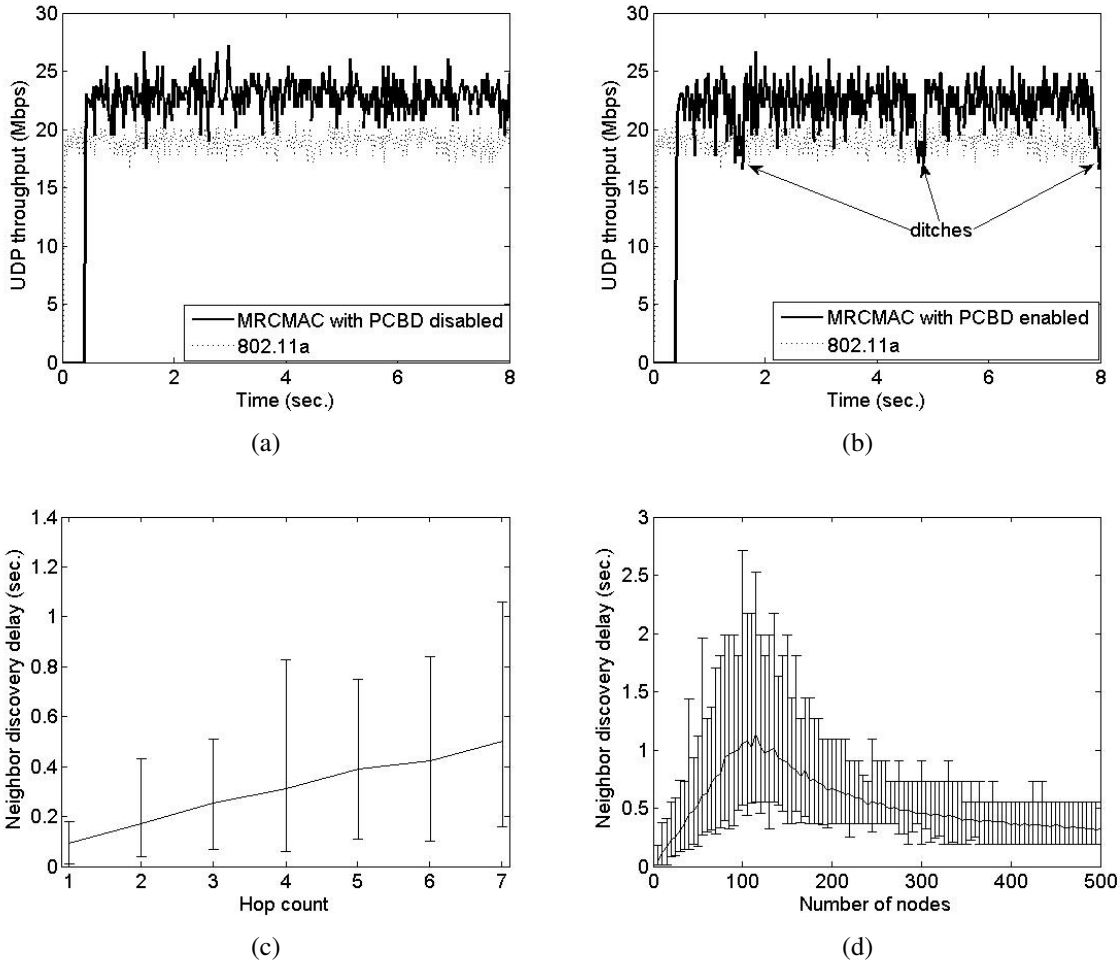


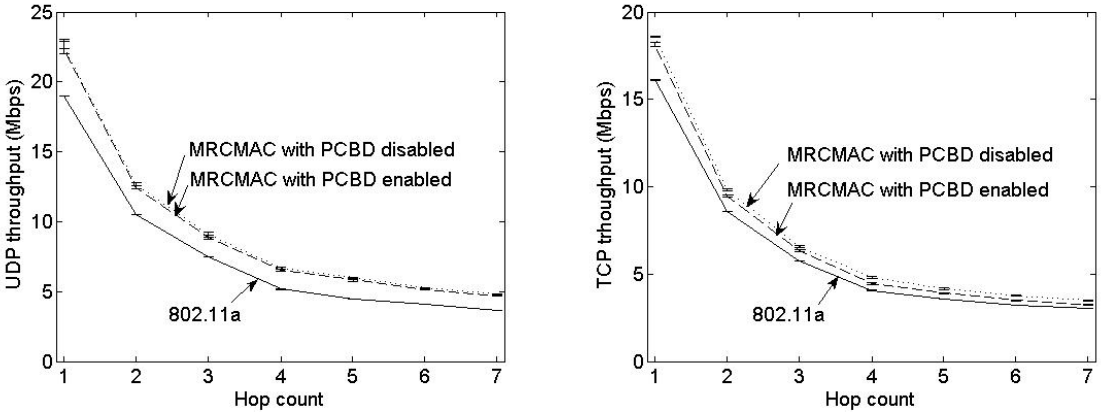
Figure 6.5: The neighbor discovery delay and single-hop throughput of MRCMAC: (a) a microscopic view of UDP throughput when PCBD disabled in MRCMAC,(b) a microscopic view of UDP throughput when PCBD enabled in MRCMAC, (c) the neighbor discovery delay in a multi-hop network, and (d) the neighbor discovery delay in a random network.

the PCBD mechanism. In Fig. 6.6(b), one can see that MRCMAC's TCP throughput gain over 802.11a's (the throughput gain is defined as $(T_{MRCMAC}/T_{802.11}) - 1$, where T_{MRCMAC} and $T_{802.11}$ are the throughputs of MRCMAC and 802.11a, respectively) becomes less than what was obtained in the UDP simulations. Meanwhile, the throughput loss caused by the PCBD option also becomes appreciable. Both changes can be explained by the increased jitter that the MRCMAC's switching delay or the PCBD mechanism causes. It is known that TCP's congestion control mechanism responds to network jitter, which adversely impacts the throughput [1].

Fig. 6.6(c) plots MRCMAC's throughput gain over 802.11a's. The figure is quite interesting because the UDP throughput gain is increasing with hop count but the TCP throughput gain is decreasing. The UDP curve can be explained by one observation made in [48]. This observation states that the UDP throughput is proportional to $(1 - \rho) \cdot data\_rate$, where ρ refers to the frame collision probability. Apparently a single-hop network has less collision than a two-hop one. More generally, an n-hop network has less collision than an $(n + 1)$-hop network. Increasing the data rate (using channel bonding) can decrease the probability of collision. If there is little collision, the throughput improvement of MRCMAC over 802.11a that is contributed by the $1 - \rho$ item is limited. In our case, therefore, when a route becomes longer and the collision probability is higher, the room for the throughput improvement also becomes greater. For the TCP curve, because of its congestion control mechanism, the TCP sender responds to the network delay and jitter caused by collision and refrains from sending enough packets to fully utilize the bandwidth [1]. This explains why the TCP's throughput decreases with the hop count.

6.4.2.3 Response to Spectrum Variability

In this experiment, we observe MRCMAC's performance under spectrum variability. Note that in addition to primary user activities, CCJ attacks could also cause spectrum variability. The network



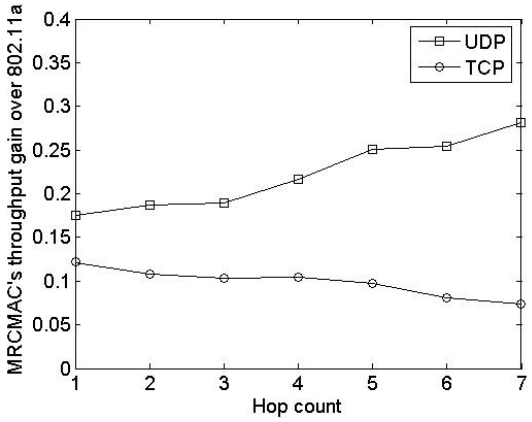


Figure 6.6: The throughput MRCMAC induces in a single-flow multi-hop network: (a) UDP throughput, (b) TCP throughput, and (c) throughput gain over 802.11a.

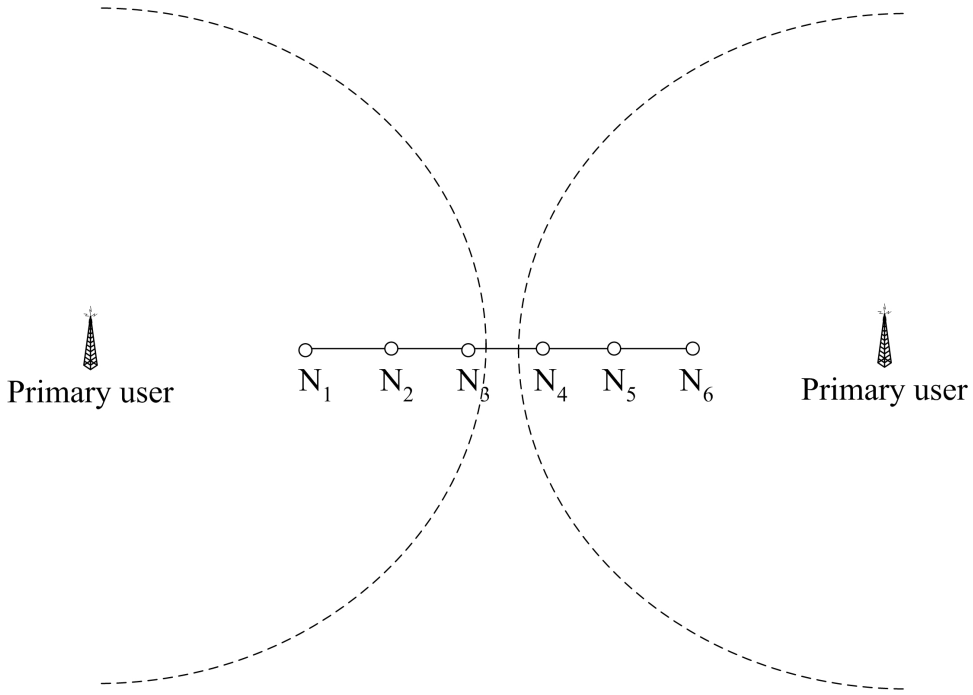


Figure 6.7: The network layout for simulating spectrum variability.

layout and channel availability used in the simulation are shown in Fig. 6.7 and Table 6.2. We put a multi-hop network with six nodes under the influence of two different primary users, which leads to spectrum heterogeneity in the network. There is a single UDP flow transmitting from N_1 to N_6. Table 6.2 shows how the channel availability changes with time, which simulates spectrum mobility in the network. The simulated spectrum mobility is relatively fast since the commonly available channels for all nodes change every ten seconds. Also, the simulated changes cover six representative scenarios where the number of channels changes between "all" and "many" (e.g., at 10s and at 60s), between "many" and "one" (e.g., at 20s and at 50s), and between "one" and "zero" (e.g., at 30s and at 40s). In spite of the very dynamic channel availability, as Fig. 6.8 shows, MRCMAC is able to fast adapt to channel variability and best utilize the existing channel opportunities. In particular, no matter how the channel availability changes, in the simulation the worst delay for MRCMAC to rebuild a stable link is less than one second. In certain cases (e.g., at time 10s and time 60s), MRCMAC can smoothly adapt to channel variability in no time.

Table 6.2: The channel availability for simulating spectrum variability.

Time	v_1, v_2, and v_3	v_4, v_5, and v_6	Available channels
0s-10s	0x1FFFF	0x1FFFF	CH0-16
10s-20s	0x01FFF	0x1FFF0	CH4-12
20s-30s	0x001FF	0x1FF00	CH7
30s-40s	0x0001F	0x1F000	N/A
40s-50s	0x001FF	0x1FF00	CH7
50s-60s	0x10FFF	0x1FFF0	CH4-12
60s-70s	0x1FFFF	0x1FFFF	CH1-16

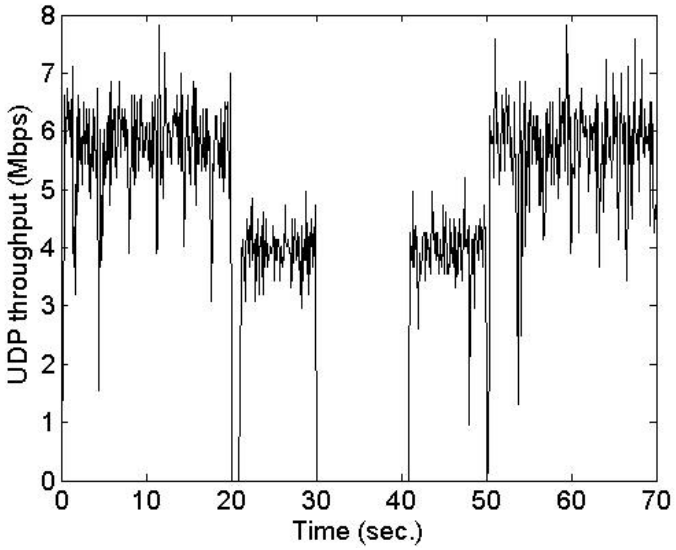


Figure 6.8: MRCMAC's UDP throughput under spectrum variability.

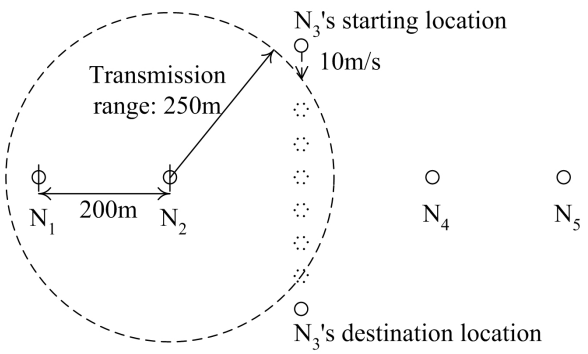


Figure 6.9: The simulation setting with node mobility.

6.4.2.4 Impact of Node Mobility

We now study the performance of MRCMAC when node mobility is considered. Since MRCMAC is designed for ad hoc CR networks, it should be able detect link breakage caused by node mobility and effectively build a new link for nodes that move to the transmission range of each other. Therefore, we conduct the simulation on both MRCMAC and 802.11a using the setting shown in Fig. 6.9. We set up a UDP flow sending from N_1 to N_5. The results in Fig. 6.10 show that MRCMAC indeed has the ability to handle node mobility. However, compared with 802.11a, MRCMAC suffers from 0.4s delay when starting building up a route for transmission. This delay is caused by the neighbor discovery delay of MRCMAC. As is shown in Fig. 6.5(c), 0.4s is a possible neighbor discovery delay of a 3-hop network[9].

6.4.2.5 Impact of Clock Drift

In MRCMAC's synchronous hopping state, nodes should ideally switch to a different (bonded) channel simultaneously. In practice, however, there exists clock drift at different nodes. In this experiment, we vary the clock drift error (denoted as e) in a multi-hop CR network and measure

[9]We look up the neighbor discovery delay of a 3-hop network instead of a 5-hop network because before N_3 enters the transmission range of N_2 and N_4, both nodes N_1,N_2 and nodes N_4,N_5 should have completed neighbor discovery much earlier.

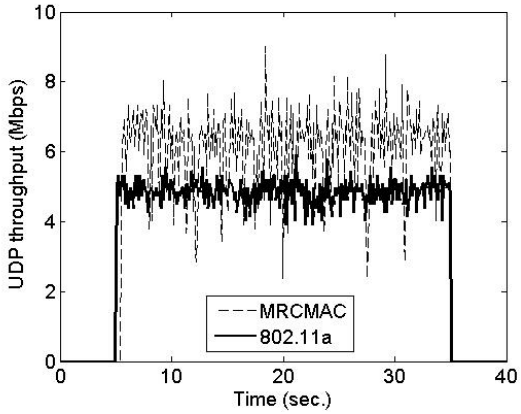


Figure 6.10: The UDP throughput as node N_3 moves.

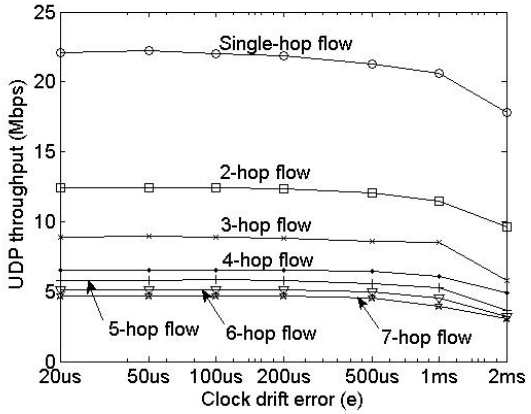


Figure 6.11: The multi-hop network UDP throughput when clock drift error is considered.

its UDP throughput. Each node in the network has a clock drift value that is randomly distributed in $[-e, e]$. Fig. 6.11 shows the simulation result averaged over ten independent simulation runs. In the simulations, MRCMAC has shown to be robust to moderate clock drift error. Specifically, when the clock drift error is 200μs, the throughput loss is less than one percent. This result justifies the adoption of protocols to conduct loose synchronization, as discussed in Section 6.3. The protocol in [19] can ensure that participating nodes have an clock drift error within a few microseconds. According to Fig. 6.11, it has negligible impact on the network throughput.

6.4.2.6 Scalability

In order to know how well MRCMAC scales with network size and complexity, we study MRCMAC's performance in multi-flow random networks. A random network is set up by placing 500 nodes randomly in a 2000m$\times$2000m area. We simulate two cases when the nodes are static or mobile. In the case where nodes are mobile, each node's movement follows the random-waypoint mobility model [4], in which we use a maximum speed of 10m/s, a minimum speed of 5m/s, and a maximum node pausing time as 10s. We run the random waypoint model for 900s before the simulation to ensure that the model enters a steady state. We randomly choose ten nodes from the network and set up five UDP flows among them. As before, ten independent simulation runs are conducted. Fig. 6.12 shows the results. In the figure, when the nodes are static, MRCMAC has an 11.6% throughput gain over 802.11a, and when the nodes are mobile, the gain drops to 3.2%. Compared to the single-flow multi-hop network throughput (see Subsection 6.4.2.2), MRCMAC in the static case now has less throughput gain. We reason it as a result of the need to exchange more CV information among all nodes. However, given the fact that the increase in the number of nodes is significant (from 7 to 500[10]), the loss in the throughput gain 13.7% (= 25.3%– 11.6%) has been acceptable. As is discussed in Section 6.2, the technique of "selective CV forwarding" has been critical to minimize the throughput loss. The case with mobile nodes results in another 8.4% loss in the throughput gain. The additional Route Request retransmission delay in DSR that MRCMAC experiences (see Subsection 6.4.2.1) could be the major cause.

[10]We compare the simulation results with that of 6-hop (7-node) single-flow simulations because the average route length in the random networks turns out to be close to six hop.

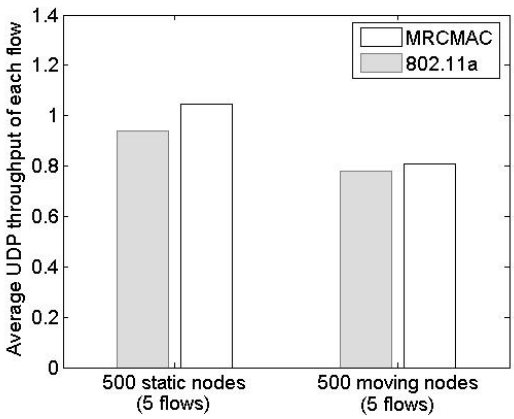


Figure 6.12: The average UDP throughput in a 5-flow random network with 500 nodes.

6.5 Chapter Summary

This chapter presented MRCMAC, a cognitive MAC that is robust against CCJ attacks. Unlike existing cognitive MAC protocols, MRCMAC no longer requires a control channel but distributes control frames to all available channels. This distinguishing feature could significantly decrease the effectiveness of CCJ attacks. In addition, MRCMAC enables fast response to spectrum variability, enables fast neighbor discovery, supports channel bonding, and supports coexistence of different CR networks. The protocol achieves these benefits by adopting a few useful techniques: pseudo-random channel hopping, selective CV forwarding, and PCBD. We used extensive simulation results to show that the performance of MRCMAC meets its design goals. Also, as the results show, MRCMAC exhibits good resilience under varying network conditions.

Chapter 7

Conclusion and Future Work

The CR technology promises to alleviate the spectrum shortage problem and bring about remarkable improvements in the efficiency of spectrum utilization. Although the operational aspects of CRs and CR networks have attracted great research interest, the research on their security aspects has been very limited. In this dissertation, we discussed a few security issues that pose serious threats to CR networks. Also, we explored possible countermeasures against the security threats. This research has provided a deep understanding of the security issues in CR networks and would have a profound impact on enhancing the attack resilience of the networks. In this chapter, we summarize the research work and highlight the major advances that have been accomplished. We also discuss future directions to extend the current work.

7.1 Research Summary

This research has discussed three potential attacks that can be launched at the physical layer or at the link layer of a CR network—PUE attacks, SSDF attacks, and CCJ attacks. In a PUE attack, a

malicious secondary tries to gain priority over other secondary users by transmitting signals that emulate the characteristics of a primary user's. In an SSDF attack, an attacker sends false local spectrum sensing results to a data collector, causing the data collector to make a wrong spectrum sensing decision. In a CCJ attack, an adversary strategically interferes with the transmission or reception of control information in a control channel. All the attacks could wreak havoc to the normal operation of CR networks.

A PUE attack belongs to a physical-layer attack. An SSDF attack is launched at the link layer (because it requires using the link layer to transmit local spectrum sensing data), but it takes effect at the physical layer. Both PUE attacks and SSDF attacks can disrupt spectrum sensing in a CR network. A CCJ attack is considered as a link-layer attack since it disrupts the MAC in a CR network. Depending on the cognitive MAC protocol being used, a CCJ attack can be launched with many options. While PUE attacks and SSDF attacks only exist in CR networks, CCJ attacks can be launched similarly in conventional wireless networks but pose a more difficult problem and require a different solution. These attacks represent great security threats that must be effectively contained in a real deployment of a CR network.

For the countermeasure against PUE attacks, the dissertation presented a transmitter verification scheme to detect the attacks (Chapter 4) . One of the distinguishing features of the proposed transmitter verification scheme is the fact that in addition to signal characteristics and energy levels, the scheme also verifies the transmitter's position in the verification process. The core of the transmitter verification scheme is a non-interactive location verification scheme or a non-interactive localization scheme. For the location verification scheme, we proposed DRT and DDT. DRT is based on the RSS of a signal source while DDT utilizes the received signal's relative phase difference when the signal is received at different receivers. Simulation results showed that several

factors, such as the location of the attacker's transmitter relative to the LVs, can impact the performance of the two location verification schemes. For the localization scheme, we proposed a localization system that uses an underlying WSN to collect RSS distributions in a CR network and uses data smoothing to pinpoint a transmitter's location. Security analysis and simulation results showed that the proposed localization scheme has been effective and robust against a number of attacks.

For the countermeasure against SSDF attacks, the dissertation explored advanced data fusion rules for attack mitigation. In particular, we used SPRT for realizing variable number of sampling and suggested WSPRT that further added a reputation-based mechanism (Chapter 5). With the features of variable number of sampling and the reputation-based mechanism, WSPRT is a data fusion technique robust against SSDF attacks. We compared all existing fusion techniques using simulation and analysis. It was shown that decision fusion techniques are easy to implement and can generate correct spectrum sensing decisions that can meet non-strict requirements. Bayesian detection, Neyman-Pearson test, SPRT, and WSPRT are more difficult to realize but can achieve better performance, particularly in terms of miss detection ratios. Among them, WSPRT is the most robust against SSDF attacks, with more system overhead as its cost.

For the countermeasure against CCJ attacks, the dissertation proposed MRCMAC for attack mitigation (Chapter 6) . Unlike existing cognitive MAC protocols, MRCMAC does not require a control channel but instead distributes control frames to all available channels. This distinguishing feature could significantly enhance a cognitive MAC protocol's resilience against CCJ attacks. The design of MRCMAC adopted several novel techniques: pseudo-random channel hopping, selective CV forwarding, and PCBD. Because of these techniques, MRCMAC enables fast response to spectrum variability, enables fast neighbor discovery, supports channel bonding, and supports

coexistence of different CR networks. We used extensive simulation results to show that the performance of MRCMAC meets its design goals. Also, as the results show, MRCMAC exhibits good resilience under varying network conditions.

The research on these potential attacks and their countermeasures represents one of the very first attempts to investigate security issues in CRs and CR networks. While large-scale deployment of CRs and CR networks still face many difficulties in their operational aspects, it is meaningful to carefully examine related security issues at a stage as early as possible. By doing so, we can minimize the possible risks induced by insecure network implementation and minimize the cost to apply necessary security measures. For these reasons, we believe that this research has practical importance for designing and deploying real-world CR systems in future.

7.2 Future Work

Since this research represents one of the very first efforts to investigate security issues in CRs and CR networks, the research in this area is far from complete. In fact, this research could bring up a much broader range of problems related to CR security.

As far as the attacks and countermeasures discussed in the dissertation are concerned, the problems have not been completely solved. For example, for the proposed transmitter verification scheme and WSPRT, our discussion has been limited to the scenario where TV systems are primary users. However, another type of primary users, Part 74 devices, are also licensed in the TV band. Future DSA applications may be extended into other bands such as those used by wireless cellular networks. These primary users are mobile and have low transmission power. It is a challenging problem to defend against PUE attacks or to apply WSPRT when these mobile, low-power primary users exist. A possible solution is to utilize the idea of the "REM" [79]. REM is an integrated

database that consists of comprehensive multi-domain information for a CR network, including the locations and activities of radio devices. Given that such information is reliable, it is possible to verify a primary transmitter or to calculate the *a priori* probabilities required by WSPRT using the location information stored in the REM. Besides REM, when we consider PUE attacks, an alternative approach could be to use the intrinsic characteristics of radio frequency (RF) signals to distinguish and identify emitters—i.e., RF fingerprinting. Obviously, RF fingerprinting would have the advantage of being able to verify a mobile, low-power primary user. Another example is the data fusion techniques used in MRCMAC. We assumed a decision fusion method with an "OR" rule. However, in Chapter 5, we knew that this fusion method is not the best in many situations. It could be a very challenging problem to adapt MRCMAC to embracing other fusion rules such as Bayesian detection, Neyman-Pearson test, SPRT, or WSPRT.

The security issues investigated in the dissertation have focused on two important mechanisms in CR networks, i.e., spectrum sensing and cognitive MAC. However, not all security issues in these two mechanisms have been discussed. For example, a full-fledged cognitive MAC protocol should also consider coexistence of multiple CR networks. However, we have not explored coexistence mechanisms in depth or discussed any related potential security problems. More importantly, in addition to spectrum sensing and cognitive MAC, there are also other mechanisms in CR networks. More security-related research is needed for those mechanisms as well. Because CR networks have different characteristics at the physical layer and the link layer compared to conventional wireless networks, CR networks also require some changes in upper layers and cross-layer design [2]. The resulting differences could become sources of latent security problems. Ideally, we should fully scrutinize these differences with a security mind.

A final note on the future work is about implementation. This research has heavily relied on computer simulations to study the behavior of CR networks and the proposed mechanisms. While this

methodology is reasonable in a certain degree, it cannot be a substitute for real implementation. In fact, from the security vulnerabilities found in 802.11 [5], one can conclude that even if the design of a protocol is intended to be secure, its implementation does not always align with the design, either because the design neglects some practical issues during the implementation or because the implementation is insecure by itself. Similarly, the proposed mechanisms in the dissertation should not be taken as is until their implementation proves they work. In this research, to ensure the credibility of the simulation work, we have taken a few indirect approaches. First, we analyze the simulation results to see if they match existing theoretic models. Second, we compare the simulation results with related experimental results that are seen in other research works. Third, we vary simulation settings to observe if the different results are consistent. However, to ultimately verify and validate the simulation work conducted in this research, implementation would be the only direct solution. That said, implementation of the research ideas in this dissertation remains a challenging task. To successfully deploy a real CR network, many operational issues have yet to be solved [2]. To further deploy the security countermeasures proposed in this research, we also need to expect major technological advances related to other areas of wireless communications.

7.3 Concluding Thoughts

This dissertation has investigated several specific security problems in CR networks. The research findings in the dissertation are helpful to enhancing attack resilience in future CR network deployment. In this final section, we discuss some of the important factors one needs to consider while applying our research findings or studying other security problems in CR networks.

The first factor to consider is policy issues. Although this research has been mainly addressing technical issues, it is actually built upon some spectrum regulation policies. Obviously, opening up

licensed bands to unlicensed operations based on the DSA paradigm has been the premise policy for the whole concept of CR networks. However, existing policies have also left some blured areas, and the choices one has in the blured areas could significantly impact technical issues. For example, we discussed PUE attacks (see Chapter 3). We think it is a threatening attack and the attacker should be punished for launching the attack. This motivation drove us to develop the technique to pinpoint the attacker that could potentially lead to his/her physical capture. However, one could also argue that since PUE attackers do not interfere with primary users, emulating primary user signals in a spectrum band is perfectly legal when the real primary user is not accessing the band. From that perspective, localizing the attacker becomes much less useful.

The second factor to consider is the tradeoffs to make while taking security measures in CR networks. In fact, such a tradeoff exists in any security problem in the real world. Specifically in this research, applying the transmitter verification scheme could detect PUE attacks, but it takes longer time for spectrum sensing taxes a CR network's capability of location verification or localization. Applying WSPRT could enhance resilience against SSDF attacks but it requires a fusion center to collect much more local spectrum sensing results. Applying MRCMAC could enhance resilience against CCJ attacks but its channel hopping is also an overhead for CR networks. These proposed schemes are only useful when the cost to implement them is overshadowed by the benefits they bring to a CR network.

The last but not the least, security measures in a CR network need to evolve along with other technological advances. In this dissertation, we have been forward-looking enough to consider some technologies that is not completely mature now but will be promising. For example, the localization technique proposed for the transmitter verification scheme (see Chapter 4) uses an underlying WSN to collect RSS measurements. We think this approach is necessary because technological advances could change the tradeoff that is to be considered while taking security measures. In

history, we have seen cases where some encryption scheme originally considered unbreakable got broken as computing speed increased. For the same reason, when applying any security measure in CR networks (e.g., exchanging secure messages among LVs (see 4.2.3.2)), one should also be forward-looking enough to set appropriate key length or, when the assumptions of a security measure no longer hold, find a more advanced solution.

Bibliography

[1] I. Aad, J. Hubaux, and E. W. Knightly, "Denial of service resilience in ad hoc networks," *Proc. ACM MobiCom*, Sep. 2004, pp. 202–215.

[2] 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, Sep. 2006, pp. 2127–2159.

[3] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," *Proc. ACM Mobicom*, Sep. 2004, pp. 216–230.

[4] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," *IEEE Trans. Mobile Computing*, Vol. 2(3) , Jul.-Sep. 2003, pp.257–269.

[5] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: real vulnerabilities and practical solutions," *Proc. 12th conference on USENIX Security Symposium*, Aug. 2003, pp. 15–28.

[6] V. Brik, E. Rozner, S. Banerjee, and P. Bahl, "DSAP: a protocol for coordinated spectrum access," *Proc. IEEE DySPAN*, Nov. 2005, pp. 611–614.

[7] M. M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans, "DIMSUMnet: new directions in wireless networking using coordinated dynamic spectrum," *Proc. Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, Jun. 2005, pp. 78–85.

[8] D. Cabric and R. W. Brodersen, "Physical layer design issues unique to cognitive radio systems," *Proc. IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2005, pp. 759–763.

[9] S. Capkun, M. Cagalj, and M. Srivastava, "Secure localization with hidden and mobile base stations," *Proc. IEEE Infocom*, Apr. 2006, pp. 1–10.

[10] S. Capkun and J.-P. Hubaux, "Secure positioning in wireless networks," *IEEE J. Selected Areas in Communications*, Vol. 24(2), Feb. 2006, pp. 221–232.

[11] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," *Proc. Thirty-Eighth Asilomar Conf. Signals, Systems and Computers*, Nov. 2004, pp. 772–776.

[12] L. Cao and H. Zheng, "Distributed spectrum allocation via local bargaining," *Proc. of IEEE SECON*, Sep. 2005, pp. 475–486.

[13] K. Challapali, S. Mangold and Z. Zhong, "Spectrum agile radio: detecting spectrum opportunities," *Proc. 6th Annual Int'l Symp. Advanced Radio Technologies*, Mar. 2004, pp. 61–68.

[14] G. Chouinard, *IEEE P802.22 Wireless RANs: Minutes of the "Channel Model" Sub-group Teleconference*, July 2005, available at: http://www.ieee802.org/22/.

[15] T. Chen, H. Zhang, G. M. Maggio, and I. Chlamtac, "CogMesh: a cluster-based cognitive radio network," *Proc. IEEE DySPAN*, Apr. 2007, pp. 168–178.

[16] C. Cordeiro, K. Challapali, D. Birru, and S. Shankar, "IEEE 802.22: the first worldwide wireless standard based on cognitive radios," *Proc. IEEE DySPAN*, Nov. 2005, pp. 328–337.

[17] K. Dogancay and D. A. Gray, "Closed-form estimators for multi-pulse TDOA localization," *Proc. 8th Int'l Symp. Signal Processing and Its Applications*, Aug. 2005, pp. 543–546.

[18] C. Doerr, M. Neufeld, J. Fifield, T. Weingart, D. C. Sicker, and D. Grunwald, "MultiMAC—an adaptive MAC framework for dynamic radio networking," *Proc. IEEE DySPAN*, Nov. 2005, pp. 548–555.

[19] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcast," *ACM SIGOPS Operating Systems Review*, Vol. 36, 2002, pp. 147–163.

[20] W. Enck, P. Traynor, P. McDaniel, T. L. Porta, "Exploiting open functionality in SMS-capable cellular networks," *Proc. ACM CCS*, Nov. 2005, pp. 393–404.

[21] Federal Communications Commission, "Facilitating opportunities for flexible, efficient, and reliable spectrum use employing spectrum agile radio technologies," *ET Docket No. 03-108*, Dec. 2003, available at: http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-03-322A1.pdf.

[22] Federal Communications Commission, "Unlicensed operation in the TV broadcast bands and additional spectrum for unlicensed devices below 900 MHz in the 3GHz band," *ET Docket No. 04-186*, May 2004, available at: http://fjallfoss.fcc.gov/edocs_public/attachmatch/DA-04-4013A1.pdf.

[23] Federal Communications Commission, "E911 requirements for IP-enabled service providers," *ET Docket No. 05-196*, May 2005, available at: http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-05-116A1.pdf.

[24] G. Ganesan and Y. Li, "Cooperative spectrum sensing in cognitive radio networks," *Proc. IEEE DySPAN*, Nov. 2005, pp. 137–143.

[25] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE J. Selected Areas in Communications*, Vol. 23(2), Feb. 2005, pp. 201–220.

[26] F. Herzel, G. Fischer, and H. Gustat, "An integrated CMOS RF synthesizer for 802.11a wireless LAN," *IEEE Journal of Solid-State Circuits*, Vol. 38(10), Oct. 2003, pp. 1767–1770.

[27] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher, "Range-free localization schemes in large scale sensor networks," *Proc. ACM MobiCom*, Sep. 2003, pp. 81–94.

[28] J. Hillenbrand, T. A. Weiss, and F. K. Jondral, "Calculation of detection and false alarm probabilities in spectrum pooling systems," *IEEE Communications Letters*, Vol. 9(4), Apr. 2005, pp. 349–351.

[29] IEEE 802.11b/D3.0, Wireless LAN Medium Access Control (MAC) and Physical (PHY) Layer Specification: High Speed Physical Layer Extensions in the 2.4GHz Band, 1999.

[30] IEEE 802 LAN/MAN Standards Committee 802.22 working group on WRANs (Wireless Regional Area Networks), 2007, available at: http://www.ieee802.org/22/.

[31] K. Jain, J. Padhye, V. N. Padmanabha, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *Proc. ACM Mobicom*, Sep. 2003, pp. 66–80.

[32] J. Jia, Q. Zhang, and X. S. Shen, "HC-MAC: a hardware-constrained cognitive MAC for efficient spectrum management," *IEEE J. Selected Areas in Communications Special Issue on Cognitive Radio Theory and Applications*, Vol. 26(1), Jan. 2008, pp. 106–117.

[33] D. B. Johnson, Y.-C. Hu, and D. A. Maltz, *The Dynamic Source Routing (DSR) Protocol for Mobile Ad Hoc Networks for IPv4 (DSR)*, RFC 4728, IETF, Feb. 2007.

[34] S. Krishnamurthy, M. Thoppian, S. Venkatesan, and R. Prakash, "Control channel based MAC-layer configuration, routing and situation awareness for cognitive radio networks," *Proc. IEEE MILCOM*, Oct. 2005, pp. 455–460.

[35] M. G. Kuhn, "An asymmetric security mechanism for navigation signals," *Proc. Information Hiding Workshop*, May 2004, pp. 239–252.

[36] Y. W. Law, P. Hartel, J. Hartog, and P. Havinga, "Link-layer jamming attacks on S-MAC," *Proc. the Second European Workshop on Wireless Sensor Networks*, Jan. 2005, pp. 217–225.

[37] Y. W. Law, L. van Hoesel, J. Doumen, P. Hartel, and P. Havinga, "Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols," *Proc. the 3rd ACM workshop on Security of ad hoc and sensor networks (SANS)*, Nov. 2005, pp. 76–88.

[38] X. Liu, G. Noubir, R. Sundaram, and S. Tan, "SPREAD: foiling smart jammers using multi-layer agility," *Proc. IEEE INFOCOM*, May 2007, pp. 2536–2540.

[39] T. Locher, R. Wattenhofer, and A. Zollinger, "Received-signal-strength-based logical positioning resilient to signal fluctuation," *Proc. 1st ACIS Int'l Workshop on Self-Assembling Wireless Sensor Networks*, May 2005, pp. 396–402.

[40] L. Lu, S.-Y. Chang, J. Zhang, L. Qian, J. Wen, V. K. N. Lau, R. S. Cheng, R. D. Murch, W. H. Mow, and K. B. Letaief, *Technology Proposal Clarifications for IEEE 802.22 WRAN Systems*, Mar. 2006, available at: http://www.ieee802.org/22/.

[41] L. Ma, X. Han, and C.-C. Shen, "Dynamic open spectrum sharing MAC protocol for wireless ad hoc networks," *Proc. IEEE DySPAN*, Nov. 2005, pp. 203–213.

[42] S. M. Mishra, A. Sahai, and R. Brodersen, "Cooperative sensing among cognitive radios," *Proc. IEEE ICC*, June 2006, pp. 1658–1663.

[43] J. Mitola, "Cognitive radio for flexible mobile multimedia communication," *Proc. IEEE International Workshop on Mobile Multimedia Communications (MoMuC)*, Nov. 1999, pp. 3–10.

[44] J. Mitola, *Cognitive radio: an integrated agent architecture for software defined radio*, PhD Dissertation, Royal Institute of Technology (KTH), Stockholm, Sweden, June 2000.

[45] J. Mo, H.-S. W. So, and J. Walrand, "Comparison of multi-channel MAC protocols," *Proc. the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWiM)*, Oct. 2005, pp. 209–218.

[46] V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein, "Using channel hopping to increase 802.11 resilience to jamming attacks," *Proc. IEEE INFOCOM*, May 2007, pp. 2526–2530.

[47] H. Nan, T.-I. Hyon, and S.-J. Yoo, "Distributed coordinated spectrum sharing MAC protocol for cognitive radio," *Proc. IEEE DySPAN*, Apr. 2007, pp. 240–249.

[48] P. C. Ng and S. C. Liew, "Throughput analysis of IEEE 802.11 multi-hop ad hoc networks," *IEEE/ACM Transaction on Networking (TON)*, Vol. 15(2), Apr. 2007, pp. 309–322.

[49] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," *Proc. IEEE GLOBECOM*, Nov. 2001, pp. 2926–2931.

[50] G. Noubir, "On connectivity in ad hoc networks under jamming using directional antennas and mobility," *Wired/Wireless Internet Communications*, vol. LNCS 2957, Springer-Verlag, 2004, pp. 186–200.

[51] The Network Simulator—ns-2, 2007, available at http://www.isi.edu/nsnam/ns/.

[52] A. Pandharipande, J.-M. Kim, D. Mazzarese, and B. Ji, *IEEE P802.22 Wireless RANs: Technology Proposal Package for IEEE 802.22*, Nov. 2005, available at: http://www.ieee802.org/22/.

[53] P. Pawelczak, R. V. Prasad, X. L. Xia, and I. G. M. M. Niemegeers, "Cognitive radio emergency networks—requirements and design," *Proc. IEEE DySPAN*, Nov. 2005, pp. 601–606.

[54] P. Papadimitratos, S. Sankaranarayanan, and A. Mishra, "A bandwidth sharing approach to improve licensed spectrum utilization," *IEEE Communications Magazine*, Vol. 43(12), Dec. 2005, pp. s10–s14.

[55] C. Peng, H. Zheng, and B. Y. Zhao, "Utilization and fairness in spectrum assignment for opportunistic spectrum access," *Mobile Networks and Applications*, Vol. 11(4), Aug. 2006, pp. 555–576.

[56] J. G. Proakis, *Digital Communications*, McGraw-Hill, 4th edition, 2000.

[57] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, 1996.

[58] C. Raman, R. D. Yates, and N. B. Mandayam, "Scheduling variable rate links via a spectrum server," *Proc. IEEE DySPAN*, Nov. 2005, pp. 110–118.

[59] J. H. Reisert, *Understanding and Using Antenna Radiation Patterns*, 2007, available at: http://www.astronwireless.com/topic-archives-antenna-radiation-patterns.asp.

[60] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*, Wiley-Interscience, Sep. 2003.

[61] T. Roos, P. Myllymaki, and H. Tirri, "A statistical modeling approach to location estimation," *IEEE Trans. Mobile Computing*, Vol. 1(1), Jan.-Mar. 2002, pp. 59–69.

[62] A. Sahai, N. Hoven, and R. Tandra, "Some fundamental limits on cognitive radio," *Proc. Allerton Conference on Communication, Control, and Computing*, Oct. 2004, pp. 1–10.

[63] S. Y. Seidel, T. S. Rappaport, S. Jain, M. Lord, and R. Singh, "Path loss, scattering and multipath delay statistics in four European cities for digital cellular and microcellular radiotelephone," *IEEE Trans. Vehicular Technology*, Vol. 40(4), Nov. 1991, pp. 721–730.

[64] S. Shankar, C. Cordeiro, and K. Challapali, "Spectrum agile radios: utilization and sensing architectures," *Proc. IEEE DySPAN*, Nov. 2005, pp. 160–169.

[65] J. S. Simonoff, *Smoothing Methods in Statistics*, Springer-Verlag, 1996.

[66] H.-S. W. So, J. Walrand, and J. Mo, "McMAC: a parallel rendezvous multi-channel MAC protocol," *Proc. IEEE WCNC*, Mar. 2007, pp. 334–339.

[67] M. Stahlberg, "Radio jamming attacks against two popular mobile networks," *Proc. Tik-110.501 Seminar on Network Security*, Helsinki University of Technology, 2000.

[68] W. Stallings, *Cryptography and Network Security: Principles and Practice*, Second Edition, Prentice-Hall, 1999.

[69] E. P. J. Tozer, *Broadcast Engineer's Reference Book*, Elsevier, 2004.

[70] G. Staple and K. Werbach, "The end of spectrum scarcity," *IEEE Spectrum*, Vol. 41(3), Mar. 2004, pp. 48–52.

[71] P. K. Varshney, *Distributed Detection and Data Fusion*, Springer-Verlag New York, 1997.

[72] M. C. Vuran and I. F. Akyildiz, "A-MAC: adaptive medium access control for next generation wireless terminals," *IEEE/ACM Transactions on Networking*, Vol. 15(3), June 2007, pp. 574–587.

[73] W. Xu, W. Trappe, Y. Zhang, and Timothy Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," *Proc. ACM MobiHoc*, May 2005, pp. 46–57.

[74] B. H. Wellenhoff, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*, Fourth edition, Springer Verlag, 1997.

[75] B. Wild and K. Ramchandran, "Detecting primary receivers for cognitive radio applications," *Proc. IEEE DySPAN*, Nov. 2005, pp. 124–130.

[76] S. A. Zekavat and X. Li, "User-central wireless system: ultimate dynamic channel allocation," *Proc. IEEE DySPAN*, Nov. 2005, pp. 82–87.

[77] H. Zheng and L. Cao, "Device-centric spectrum management," *Proc. IEEE DySPAN*, Nov. 2005, pp. 56–65.

[78] Y. Zhang, W. Liu, and W. Lou, "Anonymous communications in mobile ad hoc networks," *Proc. IEEE INFOCOM*, Mar. 2005, pp. 1940–1951.

[79] Y. Zhao, J. H. Reed, S. Mao, and K. K. Bae, "Overhead analysis for radio environment map-enabled cognitive radio networks," *Proc. IEEE Workshop on Networking Technologies for Software Defined Radio Networks*, Sep. 2006, pp. 18–25.

[80] Q. Zhao, L. Tong, and A. Swami, "Decentralized cognitive mac for dynamic spectrum access," *Proc. IEEE DySPAN*, Nov. 2005, pp. 224–232.

[81] J. Zhao, H. Zheng, and G.-H. Yang, "Distributed coordination in dynamic spectrum allocation networks," *Proc. IEEE DySPAN*, Nov. 2005, pp. 259–268.

Vita

Ruiliang Chen received his B.E. degree in Communications Engineering in 2000, and his M.E. degree in Communications and Information Systems in 2003, both from Fudan University, China. From June 2003 to July 2004 he worked as a product engineer at the Intel Shanghai Product Corporation. From August 2004 on, he has been a Ph.D. student in the Bradley Department of Electrical and Computer Engineering at Virginia Tech. His research interests include quality of service (QoS) in wireless networks, traceback and mitigation mechanisms for thwarting denial-of-service attacks, attack-resilient routing protocols for wireless ad hoc networks, and security issues in cognitive radio networks. Below is a list of peer-reviewed publications of Ruiliang Chen.

List of Publications

1. R. Chen, J.-M. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," *IEEE Infocom 2008 mini-conference*, Apr. 2008, to appear.

2. R. Chen, J.-M. Park, Y. T. Hou, and J. H. Reed, "Toward secure distributed spectrum sensing in cognitive radio networks," *IEEE Communications Magazine Special Issue on Cognitive Radio Communications*, Apr. 2008, to appear.

3. R. Chen, J.-M. Park, and J. H. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE J. Selected Areas in Communications Special Issue on Cognitive Radio Theory and Applications*, Vol. 26, No. 1, Jan. 2008, pp. 25–37.

4. R. Chen, J.-M. Park, and R. Marchany, "A divide-and-conquer strategy for thwarting distributed denial-of-service attacks," *IEEE Trans. Parallel and Distributed Systems*, Vol. 18, No. 5, May 2007, pp. 577–588.

5. R. Chen, J.-M. Park, and R. Marchany, "RIM: router interface marking for IP traceback," *Proc. IEEE Global Telecommunications Conference (GLOBECOM '06)*, Nov.-Dec. 2006, pp. 1–5.

6. R. Chen, M. Snow, J.-M. Park, M. T. Refaei, and M. Eltoweissy, "Defending against routing disruption attacks in mobile ad hoc networks," *Proc. IEEE Global Telecommunications Conference (GLOBECOM '06)*, Nov.-Dec. 2006, pp. 1–5.

7. K. Bian, J.-M. Park, and R. Chen, "Stasis trap: cross-layer stealthy attacks in wireless ad hoc networks," *Proc. IEEE Global Telecommunications Conference (GLOBECOM '06)*, Nov.-Dec. 2006, pp. 1–5.

8. R. Chen, J.-M. Park, and M. Snow, "CARE: enhancing denial-of-service resilience in mobile ad hoc networks," *Proc. 15th Int'l Conf. Computer Communications and Networks (ICCCN '06)*, Oct. 2006, pp. 5–10.

9. R. Chen and J.-M. Park, "Ensuring trustworthy spectrum sensing in cognitive radio networks," *Proc. IEEE Workshop on Networking Technologies for Software Defined Radio Networks*, Sep. 2006, pp. 110–119.

10. R. Chen and J.-M. Park, "Attack diagnosis: throttling distributed denial-of-service attacks

close to the attack sources," *Proc. 14th Int'l Conf. Computer Communications and Networks (ICCCN '05)*, Oct. 2005, pp. 275–280.

11. R. Chen, C. Dai, and C. Gao, "A routing scheme supporting QoS in MANET based on motion prediction," *Computer Engineering*, Vol. 30, No. 2, Feb. 2004, pp. 121–123 (in Chinese).

12. C. Gao, R. Chen, and X. Zhou, "A distributed call admission control scheme for QoS support in wireless ATM networks based on auto-adaptive motion prediction," *Proc. the 2002 Int'l Conf. Parallel and Distributed Processing Techniques and Applications*, June 2002, pp. 1860–1866.

13. R. Chen, K. Liu, and C. Gao, "The research on the realization of integrated services over Intranet based on H.323," *Computer Applications and Software*, Vol. 19, No. 3, Mar. 2002, pp. 42–45 (in Chinese).

14. X. Zhou, R. Chen, and C. Gao, "A novel distributed elliptical shadow algorithm for motion prediction, resource reservation and admission control in wireless cellular networks," *Proc. 2001 Future Telecommunication Conference*, Nov. 2001, pp. 408–412.

15. R. Chen, X. Zhou, F. Zheng, and C. Gao, "A distributed algorithm enhancing the connection quality and bandwidth utilization in wireless cellular networks," *Proc. 2001 Int'l Conf. Computer Networks and Mobile Computing (ICCNMC '01)*, Oct. 2001, pp. 79–84.

16. X. Zhou, R. Chen, and C. Gao, "An elliptical shadow algorithm for motion prediction and resource reservation in wireless cellular networks," *Proc. 10th Int'l Conf. Computer Communications and Networks (ICCCN '01)*, Oct. 2001, pp. 234–239.