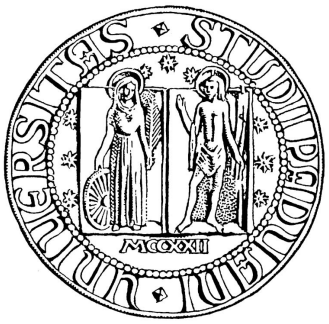
WIRELESS DEVICE IDENTIFICATION FROM A PHASE NOISE PROSPECTIVE

BY

RICCARDO RUBINO



University of Padova
Faculty of Engineering

Master of Science in Telecommunication Engineering
Department of Information Engineering

Advisor: Michele Zorzi
Co-Advisor: Kui Ren

Padova, Italy
March 2010

ACKNOWLEDGMENT

First of all, I would like to thank my Professors, Dr. Kui Ren and Dr. Michele Zorzi, for their valuable guidance and encouragement. They guided me through out my research and advised me when I needed it the most. I am also extremely grateful to Dr. Brik of University of Wisconsin. His collaboration has been essential for the successful completion of my research work. Special thanks also to all my graduate friends, especially Iñigo and Raphael, whose I have shared more than an apartment with. Without them, the stressful working days and long nights would not have brought such good memories.

I would also like to express my gratitude to my best friends Annalisa and Federico, for their paramount friendship and support through out this year. Finally, I am grateful to my mother, father and sister, for their unconditional support and understanding through out this endeavor. Without their presence, guidance, and encouragement I would not have been able to achieve my goals. They shall always be my role models and I will always cherish their love and support.

Riccardo Rubino

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Symbol	Definition
AAC	Automatic Amplitude Control
AER	Average Error Rate
AP	Access Point
AWGN	Additive White Gaussian Noise
BER	Bit Error Ratio
BW	Bandwidth
dBc	dB below the carrier
DSB	Double-sideband
DWT	Discrete Wavelet Transform
EVM	Error Vector Magnitude
FRR	False Reject Rate
IDS	Intrusion Detection Systems
I/Q	In-phase/Quadrature subcarriers
ISM	Industrial, Scientific and Medical
kNN	k-Nearest-Neighbor
MAC	Media Access Control
NIC	Network Interface Controller
NIST	National Institute for Standards and Technologies
PARADIS	Passive RAdiometric Device Identification System
PN	Phase Noise

PPM	Parts Per Million
PSD	Power-Spectral Density
QPSK	Quadrature Phase-Shift Keying
RF	Radio Frequency
RFF	Radio Frequency Fingerprinting
RMS	Root-Mean-Square
RSSI	Report Signal Strength Indication
SNR	Signal to Noise Ratio
SSB	Single-sideband
VCO	Voltage Controlled Oscillator
WCS	Worst-Case Similarity
WIS	Wireless Identification System
WLAN	Wireless Local Area Network
WWAN	Wireless Wide Area Network
$\varphi(t)$	Deterministic and random phase noise
$\phi(t)$	Random phase noise
f_m	Offset from the carrier
$h_{Tx}(t)$	Real-valued finite energy baseband pulse
l_1	Manhattan distance
$r_n(t)$	n-th received symbol
$s_n(t)$	n-th transmitted symbol
$\pounds(f)$	SSB phase noise/carrier

$V(t)$	Sinusoidal waveform produced by the synthesizer
$S_\phi(f)$	Spectral density of phase fluctuations $\phi(t)$
$S_V(f)$	Power spectrum density of $V(t)$

ABSTRACT

As wireless devices become increasingly pervasive and essential, they are becoming both a target for attacks and the very weapon with which such an attack can be carried out. Wireless networks have to face new kinds of intrusion that had not been considered previously because they are linked to the open nature of wireless networks. In particular, device identity management and intrusion detection are two of the most significant challenges in any network security solution but they are paramount for any wireless local area networks (WLANs) because of the inherent non-exclusivity of the transmission medium.

The physical layer of 802.11-based wireless communication does not offer security guarantee because any electromagnetic signal transmitted can be monitored, captured, and analyzed by any sufficiently motivated and equipped adversary within the 802.11 device's transmission range. What is required is a form of identification that is nonmalleable (cannot be spoofed easily).

For this reason we have decided to focus on physical characteristics of the network interface card (NIC) to distinguish between different wireless users because it can provide an additional layer of security. The unique properties of the wireless medium are extremely useful to get an additional set of information that can be used to extend and enhance traditional security mechanisms. This approach is commonly referred to as radio frequency fingerprinting (RFF), i.e., determining specific characteristics (fingerprint) of a network device component.

More precisely, our main goal is to prove the feasibility of exploiting phase noise in oscillators for fingerprinting design and overcome existing limitations of conventional approaches. The intuition behind our design is that the autonomous nature of oscillators among noisy physical systems makes them unique in their response to perturbations and none of the previous work has ever tried to take advantage of this.

CHAPTER 1

PRELIMINARIES

1.1 Introduction

In the last ten years we have been witnessed to the evolution brought about by wireless communications. Two important aspects have characterized this transformation: the unprecedented growth in the number of wireless users, applications and network technologies, and interoperability between these technologies, which has became a fundamental requirement for all end-user applications, like e-commerce, audio/video streaming and other services. The continuous improvements and technological advancements in wireless devices and protocols, simplification of installation and maintenance procedures as well as reduction in costs, are only few of the reasons that have declared the success of wireless all over the world.

In every market, wireless technologies are eroding more and more shares of "fixed" or "wired" networks. Actually, the most popular and widely adopted wireless network technology so far has been IEEE 802.11 networking protocol, which has been developed and improved constantly since its first appearance in 1999.

Now the 802.11x family consists of six different versions [Wikipedia, c], the most common of which are 802.11a, 802.11b, and 802.11g standard rectifications. The incredible price decrease and popularity of 802.11 capable hardware (especially with 802.11b/g) has made wireless networks cheap and easy to setup in every kind of setting, such as homes, offices, or hot spots. Because of this, the 802.11x family is the favorite protocol for wireless communication, with the exception of telephony network [Gast & Loukides, 2002].

In addition, this level of growth is expected to rise in the next few years due to the next generation of 802.11 standards, called 801.11n, which is expected to be

ready in 2010 and promises a significant increase in the maximum data rate from 54 Mbit/s to a maximum of 600 Mbit/s [Wikipedia, d].

In addition to that, the development of new important initiatives have been accelerated during the last couple of years and promise to maintain the exponential growth of this market. For example, key developments in information and communication fields include inter-machine communication (e.g. cars, household and office components equipped with a wireless interface), packet-oriented wireless systems and heterogeneous wireless infrastructures, as identified by Bria et al. [Bria et al., 2001].

1.2 Motivation

The new trend manufacturers' goal is now to promote the concept of ubiquitous computing, where users will have access to network applications and resources anytime and from anywhere. The problem is that since wireless devices have become highly pervasive and essential, they have become a soft target for attacks and also the very instruments with which these attacks can be carried out. Because of this, security issues become crucial and must be re-designed to face new threats this scenario poses. We have to keep in mind that confidentiality of information is critical for supporting applications, including on-line banking and electronic payments, e.g. e-commerce.

Security has always been one of the main concerns for networks, and all the computer and network security techniques developed for wired networks can, and must, be used to fight attacks also in a wireless scenario. Unfortunately, wireless networks have to face with new kinds of intrusion that have not been considered previously because they are linked to the open nature of wireless networks. It is clear how easy is, for an attacker, to dispose and modify wireless network devices due to their low cost and large availability. In addition, an intruder can launch an attack without the need of a physical connection, for example from another room,

building or even further. Several groups of students have already demonstrated how inadequate are the security techniques for wired networks applied to a wireless one [Borisov et al., 2001, Arbaugh et al., 2002, Walker et al., 2000].

The question arising is why can we not merely adapt methods from wire-line security? We cannot because wireless networks lack appropriate security infrastructure. As infrastructure-based WLANs represent an extension to wired LANs, the need for effective access control to network resources is equally paramount. But unlike coaxial cables or fiber, the wireless medium cannot be locked into a secure facility, and there is not the possibility to filter the packets through routers or switches based on the incoming port, as usual in a LAN where each incoming port is mapped to a single Ethernet jack in the wall. So, rogue transmitters can, without any obstructions, inject packets into the network from great distances. When radio frequency packets, transmitted by a wireless node, reach wireless access points (APs), they are converted in Ethernet or other traditional wire-line packets. After that point, there is no difference between the packets originated that by a wireless node and a fixed one, and are completely legitimate to traverse the wired infrastructure. Since device unique identifiers such as IDs and MAC addresses are easy to forge, administrators need other mechanisms to identity the source of frames within their networks.

Device identity management and intrusion detection, which are very close issues, are two of the most significant challenges in any network security solution but, in particular, for wireless local area networks (WLANs) because of the inherent non-exclusivity of the transmission medium. To overcome these hurdles, 802.11 WLAN administrators rely on intrusion detection systems (IDSs) and various cryptographic mechanisms for wireless device identity management and access control. Conventional wired IDSs cannot be simply applied to wireless networks; since they work at the data link layer or above, and they are unable to distinguish between

packets originating from legitimate wireless nodes and packets from an intruder who forges the identity of a legitimate node. Usually, they implement data collection mechanisms (host-based, network-based), or detection techniques (anomaly-based, signature-based, specification-based) but no obvious implementation is possible in wireless systems which are characterized by impossibility to rely on a centralized server, unavailability of key traffic concentration points, difficulty to secure signature distribution, and possible presence of rogue hosts. Lim et al. [Lim et al., 2003] provide a good summary of conventional IDS products.

Although enabling wireless technologies like WTLS (Wireless Transport Layered Security) within WAP (Wireless Application Protocol), WEP (Wired Equivalent Privacy), TKIP (Temporal Key Integrity Protocol), Counter Mode CBC-MAC, Wireless PKI, Smart Cards, offer security with various degrees of success and are essential to securing wireless networks and to protect the identities of the communication endpoints, the usefulness of such cryptographic schemes is, sometimes, impacted by various challenges of key distribution and, in particular, by robustly detecting and revoking compromised keys. We also want to underline that these techniques do not directly exploit the physical properties of the wireless domain to address security threats.

The unique properties of the wireless medium are extremely useful to get an additional set of information that can be used to extend and enhance traditional security mechanisms. Summarizing, the physical layer of 802.11-based wireless communication does not offer security guarantee because any electromagnetic signal transmitted can be monitored, captured, and analyzed by any sufficiently motivated and equipped adversary within the 802.11 device's transmission range. What is required is a form of identification that is nonmalleable (cannot be spoofed easily). For this reason we have decided to focus on physical characteristics of the network interface card (NIC)

to distinguish between different wireless users because it can provide an additional layer of security.

1.3 Research Objectives

As we have seen, the need of a form of identification, that is not easily spoofable, is critical for wireless systems. For this reason, we have decided to focus on physical characteristics of the NIC to discriminate between different wireless users/devices. This approach is commonly referred to as radio frequency fingerprinting, i.e., determining specific characteristics (fingerprint/signature) of a network device component with or without its cooperation. In our scenario, the fingerprinter passively observes the traffic from a targeted device (fingerprintee) in order to find unique characteristics that uniquely distinguish the wireless device.

The main objective of this thesis is to demonstrate the feasibility to exploit phase noise, that is a distinctive feature of every wireless transceiver, for wireless identification. We also analyze and propose several applications that can benefit from physical device identification and we explain how a wireless identification system (WIS) can be used to face several kinds of attacks.

1.4 Approach

Our goal is to discriminate between different wireless devices or, in other words, perform wireless identification. The need of a form of identification that is nonmalleable originates from the fact that standard identifiers are easily spoofable in a wireless system. A possible solution of this need is to identify a wireless user based on physical characteristics of his hardware. In fact, wireless identification is possible because minor variations in analog hardware of transmitters are manifested as idiosyncratic artifacts in their emitted signals and thus can be used to identify a signal's device-of-origin. These intrinsic features, also called RF impairments,

are derived from properties of the modulator, carrier frequency, power-on/power-off transients, and antenna characteristics. They are highly dependent on the wireless transceiver manufacturer's specification because the communication standards allow different implementations (e.g. the numerous realizations of the IEEE 802.11b standard [IEEE Standards Ass., 1999]). These impairments will vary from one transceiver unit to the next due to limitations of the manufacturing process. An identifier based on physical characteristics of the hardware is referred as radio frequency fingerprint (RFF) and it cannot be simply modified by their users. RFF forging typically requires modification of the device hardware and is challenging even for attackers with access to sophisticated instrumentation.

RFF is a widely used term in literature with many different meanings. The term RF fingerprinting, in general, refers to the process of classifying transmissions based on observed features of an RF signal. We can broadly classify RF features of a signal into (i) channel-specific features: that characterize the properties of the wireless channel and environment, and (ii) transmitter-specific features: that characterize the wireless transmitter itself, and are independent of the channel between the transmitter and the receiver. We have decided to ignore channel-specific features, such as channel impulse response, and to utilize only transmitter-specific features to help uniquely identify the specific wireless transmitter.

This is not a brand new idea, as we will see in Chapter 2, many other research groups have tried to perform wireless recognition based on the study of the transient in the waveform domain. Although this approach has been deeply studied, it presents several difficulties without obvious solutions, as possible to comprehend by the amount of literature of the subject [Ellis & Serinken, 2001, Hall, 2004, Hall et al., 2005, Hall et al., 2003, Shaw & Kinsner, 1989, Ureten & Serinken, 1999, Hall, 2006, Ureten & Serinken, 2007, Hippenstiel & Payal, 1996] .

Very recently, Brik et al. [Brik et al., 2008] published a brand new approach, called PARADIS, totally different from all the previous ones. First of all, they completely ignore the transient analysis considering it to be unreliable and, second, they work in the modulation domain. Instead of trying to capture the effects of RF hardware impairments in the waveform domain, they generate the signature of a network card averaging different kinds of modulation errors for a certain number of 802.11 frames. Since modulation errors and measurements are well-known and easy to perform with standard tools implemented in every receiver, this approach leads to more reliable results than the transient approach, which instead is a very brief radio emission in the order of $2\mu s$ and, because of which, is not easy to handle at all.

Our aim is to improve the PARADIS performance measuring a RF impairment ignored in all the previous studies. We extend this approach adding a new metric based on the phase noise, which is an important characteristic of oscillators, and is fundamental to every transceiver. Our simulations aim to demonstrate the feasibility of this phase noise analysis for wireless identification systems.

In brief, the phase noise is the frequency domain representation of rapid, short-term, random fluctuations in the phase of a wave, caused by time domain instabilities ("jitter") [Wikipedia, f]. All real oscillators have phase modulated noise components that spread the power of a signal to adjacent frequencies. We will discuss more in detail about phase noise (PN) in Chapter 3.

CHAPTER 2

RELATED WORK

Wireless device identification is not a brand new idea. Since the 1960s the american army has developed several systems to discriminate between friendly and enemy military airplanes [Talbot et al., 2003] or ship radars [Langley, 1993]. Consequently, all the research work is still a secret, details of the implementations are not available and we do not know if they were ever successfully used during military operations [Barton & Leonov, 1997].

Since then, especially with the proliferation of wireless devices like mobile phones, PDAs and 802.11 network cards enormously accelerated the research initiatives to detect: (i) illegally operated radio transmitters [Hippenstiel & Payal, 1995], [Toonstra & Kinsner, 1996], [Toonstra & Kinsner, 1996] (ii) device cloning [Kaplan & Stanhope, 1999], (iii) defective transmission devices [Wang et al., 2005] and (iv) identify wireless devices [Rasmussen, 2007], [Hall, 2004], [Ureten & Serinken, 2007], [Tekbas et al., 2004] by using physical characteristics of the transmitted signals [Ellis & Serinken, 2001].

A comprehensive overview of high-level issues in the context of transmitter identification is presented by Talbot et al. [Talbot et al., 2003] and by Riezenman [Riezenman, 2000]. Below, we present the most relevant work to ours in terms of signal similarities, features and objectives.

In addition to wireless identification, which is the object of our research, we will provide a summary of techniques for location distinction because, under some assumptions, they can perform wireless identification and can be used by our approach to select frames originated by a wireless node, which is one of the assumptions of our research as we will see in Section 4.7. It is possible to classify all the different

techniques based on the objective domain chosen for the analysis. We have identified three areas: (i) software-based fingerprint (ii) channel-impulse-response fingerprint (iii) radio frequency fingerprint. In the next sections we will give some examples of related works for each area.

2.1 Software-based fingerprinting

Multiple efforts have addressed the issue of distinguishing network nodes based on differences in software configuration. Unlike different approaches, here identity is based on properties of the hardware. However, users can still interfere with identification using software means. For example, one approach bases its identification on time stamps but they could be altered or disabled altogether without the need of changes in the hardware. Many applications that, for example, determine the version of a node's operating systems have already established their place in the toolkit of network administrators [Lyon,]. Typically such tools are used to identify computers running vulnerable software.

2.1.1 Identifying MAC Spoofing by Detecting Sequence Number Anomalies.

In the context of IEEE 802.11 devices, many research groups, for example [Guo & Chiueh, 2006], [Wright, 2003] and [Dasgupta et al., 2003], have discussed approaches to detect presence of multiple 802.11 devices with masqueraded MAC addresses using analysis of frame sequence numbers.

The basis for these techniques, for example for Guo et al., is that the 12-bit sequence number field in IEEE 802.11 frames increments by one for each frame generated. A wrap around will occur after 4095, the maximum frame number. Hence, by keeping track of MAC addresses to its corresponding sequence number at any point of time, it becomes possible to approximately detect an illegitimate client station with a spoofed MAC. But frames that are lost or retransmitted will affect the tracking of

802.11 frame sequence numbers. Although the approach works well for the temporal identification of unique wireless client stations, it requires the constant tracking of sequence number growth for all clients in the network. It is also susceptible to sequence number spoofing attacks, and given the small 12-bit sequence number space, the probability of sequence number collision increases with the number of wireless client stations in the network.

More in depth, Wright [Wright, 2003] proposed a threshold approach; it is solely based on sequence number gap and when it exceeds a certain threshold, a spoofing alert is raised. This algorithm is quite simplistic and tends to introduce more false positives and false negatives.

Another technique in this category has been proposed by Dasgupta et al. [Dasgupta et al., 2003]. The algorithm they proposed is composed by two phases and it is self-adaptive. First, it collects sequence numbers traces during spoofing attacks. After, the genetic algorithm based on the set of training data, auto-creates and evolves fuzzy rules to classify correctly and, moreover, protect the network from this attack. Using fuzzy logic presumably could better accommodate fluctuations in sequence number changes. However, it is not clear that this fuzzy logic approach can actually accommodate sequence number changes due to lost frames, duplicated frames, and out-of-order frames.

2.1.2 Device driver fingerprinting. This technique rises from the fact that vulnerabilities are often firmware and driver specific and, moreover, this information may be used by a hacker to launch a directed attack based on that vulnerability. As it is exposed by Franklin [Franklin et al., 2006], in this scenario the goal is to identify a user based on the differences of the drivers and firmware used by wireless NIC instead of physical machines. In particular, the analysis target is the algorithm used to scan for AP, called active scanning process, because it is not explicitly defined

in the 802.11 standard [IEEE Standards Ass., , IEEE Standards Ass.,]. Moreover differences in implementations of the active scanning process are the basis for the, so called, device driver fingerprinting.

As described in detail in the paper, during the active scanning process the client cycles through each channel in the 2.4 GHz ISM (industrial, scientific and medical) band to broadcast probe request frames. After each frame is broadcasted in a channel, it waits for a period of time defined as the *Min-Channel-Time* if the channel is idle; otherwise it waits for *Max-Channel-Time* before scanning the next channel. It is, basically, a time-series analysis; the proposed algorithm passively monitors the network and collects information regarding the time intervals between the different probe request frames. The comparison between the analyzed time series and a database of previously analyzed driver's time behaviors allow to perform driver identification. The final decision can be taken, as Franklin does in his paper, through a Bayesian classifier. Using these characteristics, wireless drivers can be determined with acceptable accuracy.

It is interesting to notice that this approach is totally passive and does not require cooperation. Unfortunately, since it is based on software behavior, it is also easily deceivable by a motivated attacker. Desmond et al. [Desmond et al., 2008] have discovered an additional drawback of this approach: there are, indeed, consistent minute differences in timing intervals of probe request frames emitted from different machines, even when they used identical NIC drivers with the result of a less accurate identification. They proposed a similar technique that mitigates this problem: it is still based on the time-analysis of probe request frames during the active scanning process, but with a generalized approach. In fact, they distinguish between unique combinations of the tuple [Machine, NIC Driver, Operating System] instead of just between unique drivers.

2.1.3 Clock Skew Deviations as Fingerprints. Kohno et al. [Kohno et al., 2005] showed that devices could be fingerprinted using the clock skew, i.e. slight drifts in devices' TCP and ICMP timestamps. As described in [Paxson, 1998], it has been found that minute deviations between the clock oscillators of different machines could result in clock skews. This is the basic idea at the base of this fingerprint technique used previously in wired networks. Kohno proposes to measure time skews based on the analysis of TCP headers with timestamps. Alternatively it is possible to send ICMP timestamp request to the wireless target. It is so possible to calculate the clock skew comparing the inter-packet arrival interval of the local and the remote node. The clock skew then becomes the fingerprint that uniquely identifies the remote fingerprintee machine. The main advantage of this technique is that it relies on the clock skews between machines, which are relatively stable and do not deviate much over time [Moon et al., 1999].

Unlike the previous approaches, here identity is based on properties of hardware. However, we have decided to classify this approach as software-based fingerprinting because attackers can still interfere with identification using software means. For example, both the TCP timestamp option and response to ICMP timestamp requests can be altered or disabled and even on Windows machines, TCP timestamps are not enabled by default.

2.2 Channel-impulse-response fingerprinting

The main goal of location distinction techniques is to discriminate transmitters based on their physical positions. These algorithms, under the assumptions that nodes remain active and do not move, perform also wireless identification. They achieve their objective analyzing and exploiting the differences in the channel response for different wireless node. Nevertheless, they lack the ability to actually make an identification or recognize a previously seen device that moved or remains silent for

some time.

In our case, we are interested to them because they provide a really useful mean to allow our approach on groups of frames guaranteed to be from the same transmitter, without having to rely on unreliable MAC addresses.

2.2.1 Location Fingerprinting through RSSI levels. Faria and Cheriton [Faria & Cheriton, 2006] propose a technique to detect spoofing attacks using a signalprint, which is a vector of median RSS measurements for a MAC address calculated at multiple APs. This approach uses multiple measurements at different receivers because RSS measurements vary due to small-scale and frequency-selective fading. The signalprint of each wireless client is constantly tracked and updated to follow the evolution of nonstationary clients, and they use the differential signal strength, i.e. the difference between a median RSS at one AP and the maximum median sensed by all APs for this MAC address, to eliminate the effects of transmission power.

They propose that two given signalprints represent two transmitters, if the median RSS values measured by at least one AP differ by 10 dB or more. Spoofing is detected when significantly different signalprints are simultaneously detected for the same MAC address, indicating the presence of distinct wireless clients transmitting from different locations. They reported a detection accuracy above 95% but a immediate comparison is not possible because the false positive rate is not reported. The authors themselves noted that RSSI methods have limitations in their ability to resolve entities On the other hand, methods that employ the characteristics of the radio frequency (RF) channel are physically limited in their resolution by the wavelength of the wireless technology, which can be on the order of tens of centimeters for IEEE 802.11b/g.

It is important to notice that in network security applications, adversaries

can easily spoof their signalprint using array antennas which send different signal strengths in the directions of different access points. In addition, similar to the previous technique in Section 2.1.1 based on sequence number tracking [Wright, 2003, Guo & Chiueh, 2006, Li & Trappe, 2007], the fingerprints are temporal in nature and not fundamentally linked to the identity of clients. Thus if a legitimate client changes its location while it is switched off, its legitimacy will be difficult to truly establish thereafter.

2.2.2 Multipath-related measurements fingerprinting. The physical layer approach to location fingerprinting was proposed in [Patwari & Kasera, 2007] and [Xiao et al., 2007] because the transmission medium between any two points is distinctive in space, time and frequency characteristics in a way that is location-specific.

In particular, Patwari et al. [Patwari & Kasera, 2007] proposed a robust location distinction mechanism that uses a physical layer characteristic of the radio channel between a transmitter and a receiver, called temporal link signature. The temporal link signature is the sum of the effects of the multiple paths from the transmitter to the receiver, each with its own time delay and complex amplitude. Such a signature changes when the transmitter or receiver changes position because the multi-path in the link change with the positions of the endpoints of that radio link. In contrast to existing techniques, location distinction using temporal link signatures does not require continuous operation, i.e. a sensor can schedule sleep, and a wireless network can send packets intermittently. When awakened from sleep or upon reception of the subsequent packet, a receiver can detect that a neighboring transmitter has moved since its past transmission.

Unlike the RSS-based technique in [Faria & Cheriton, 2006], temporal link signatures can be measured at a single receiver and require no additional complexity at the transmitter, which keeps tag cost and energy consumption low. Taking this idea

further, this physical characteristic can be further exploited to support secret key dissemination through probabilistic encryption [Xiao et al., 2007]. Thus in applications where wireless nodes rarely alter their positions, it is possible to identify them through distinctive channel signatures.

2.3 Radio Frequency Fingerprint

In the last decade, the level of interest in RFF continues to rise, initially motivated in the 1960s by the military to track the movement of enemy troops or to distinguish friendly radars from those of the enemy [Langley, 1993, Talbot et al., 2003] and recently by the need to identify malfunctioning or illegally operated radio transmitters, for example in mobile cellular carriers (e.g. Bell Nynex) to combat cloning fraud [Riezenman, 2000], in support of radio spectrum management practices or for intruder detection in wireless networks.

Inspired by these research works, attempts to measure transmitter radiative signatures from common wireless cards based on unavoidable and random fabrication differences. All the researches aim to define and measure some unique hardware properties of the wireless mobile node but it is not still clear what are the best characteristics that uniquely identify it. When one or more features are extracted from the transmitted signal of the target device, they are elaborated and represents a fingerprint of the transceiver, or otherwise called transceiverprint/signalprint/signature.

The transceiverprint is, in turn, classified as belonging to one of the profiled transceivers. After a training phase, where all the signatures are recorded, it is possible to perform identification or intruder detection through an identification phase that compares, usually with a probabilistic model, the received signature with those present in the training database.

The majority of the work in literature focuses all the analysis on the transient,

which is a very brief radio emission in the order of $2\mu s$ and, because of that, not easy to handle. In this section we will provide a survey of several works. Since it has been fundamental for our study because we used it as our framework, we will discuss a brand new approach, called PARADIS, in Section 4.2. It is particularly interesting because it completely ignores the transient, considered as unreliable. To sum up, this section provides a brief overview of the various research initiatives that have been undertaken in the area of RFF.

2.3.1 Transient approach. By far the most researched type of radiometric identification deals with the so-called signal transients. A transient is a brief radio emission, of the order of $2\mu s$ in 802.11 [IEEE Standards Ass.,], [IEEE Standards Ass., 1999], produced while the power output of an RF amplifier goes from idle to the level required for data communication. Its shape depends on the circuitry of the transmitter such as the modulator, carrier frequency, power-on requirements, and antenna characteristics.

The nature of transients is such that they are difficult to detect and there is no obvious correct way to succinctly describe them. For example, consider conversion of a transient waveform to a compact representation suitable to be input of an identification algorithm. The factors that determine a transient's shape are poorly understood, perhaps because they are of limited use, since they are very brief, they cannot serve a protocol function. Therefore, transients have to be treated essentially as arbitrary waveforms, and finding a compact representation that will be effective for all possible transients and involves heuristics and guesswork.

The amount of literature of the subject, that in part we are going to summarize in the following section, is a confirmation of this fact. These works study various techniques related to transient detection, data processing and machine learning.

2.3.1.1 Ellis and Serinken. In this work [Ellis & Serinken, 2001], Ellis and Serinken analyze the amplitude and phase captured by several receivers with the intention to understand if it is a possible approach for wireless identification. They examine the degree of variability associated to these features among a set of transceivers that included some from the same manufacturer and, moreover, likely to be more similar. In the conclusion they affirm that all the transceivers have features that is possible to measure to discriminate different devices (they consider only features derived from phase and amplitude components), unfortunately they are not necessarily unique. They also noticed that the characteristics of the fingerprints are also likely to change as a result of the environmental and the specific work situation, for instance due to temperature variation, Doppler shift, multi-path, fading, battery condition and aging.

2.3.1.2 Hall et al.. Hall et al. [Hall, 2004, Hall et al., 2005] explored a combination of features such as amplitude, phase, in-phase, quadrature, power and DWT of the transient signal. One of the biggest and fundamental issue for a transient approach is is the detection of the transient. Although the deep research they have done in [Hall et al., 2003], this problem necessitates more improvements to achieve a good accuracy, as we can see in the amount of works on this subject [Shaw & Kinsner, 1989, Ureten & Serinken, 1999].

Hall et al. tested 30 IEEE 802.11b transceivers from 6 different manufacturers and scored a classification error rate of 5.5%. Further work on 10 Bluetooth transceivers from 3 manufacturers recorded a classification error rate of 7% [Hall, 2006].

2.3.1.3 Ureten and Serinken. Ureten and Serinken [Ureten & Serinken, 2007] extracted the envelope of the instantaneous amplitude by using the Hilbert transformation and classified the signals using a Probabilistic Neural Network. The method

was tested on 8 IEEE 802.11b transceivers from 8 different manufacturers and registered a classification error rate of 2%-4%. Unfortunately, we have to underline that devices produced by different manufactures are easier to distinguish because of differences in the manufacturing process. For this reason an attacker can easily overcome an intrusion detection system based on this approach by using a device from the same manufacturer.

2.3.1.4 Hippenstiel. An interesting and unique approach is adopted by Hippenstiel [Hippenstiel & Payal, 1996]. The frequency component, coefficients of the Discrete Wavelet Transform (DWT) [Wikipedia, a], of the transient is first obtained using the Daubechies Polynomial of order 8, a wavelet filter typically used in DWT. Since the maxima of the modulus of the wavelet coefficients contains approximately the same amount of information as the transient, the wavelet coefficients at each scale are replaced by their extrema. Using this reduced representation of the coefficients (transceiverprint), an Euclidean distance measure between a given transceiverprint and the templates in each of the different classes (transceivers) is used as part of the technique.

2.3.2 Remley. Perhaps the most intuitive and straightforward way of radiometric analysis was studied by Remley [Remley et al., 2005], who visually identified differences between signals of different 802.11 transmitters. Inspired by the military technique to distinguish friendly radars from those of the enemy, called SEI [Langley, 1993], recent work, such as the work done at the National Institute for Standards and Technologies (NIST) [Remley et al., 2005], attempts to measure transmitter radiative signatures from common wireless cards based on unavoidable and random fabrication differences.

They observe that there are quantifiable differences both in the time and frequency domain between different wireless cards (even from the same manufacturer).

However, the variability of the quantifiable differences depending on the physical location and orientation of the transmitter within an environment (due to multi-path profiles, orientation of the antennas and the like) makes it harder to apply this technique in a field system.

CHAPTER 3

PHASE NOISE

An essential component in every transceiver is the voltage controlled oscillator (VCO), which is an electronic oscillator designed to be controlled in oscillation frequency by a voltage input. Unfortunately, no oscillator is perfectly stable. Due to random phase fluctuations, in the frequency domain a signal is no longer a discrete spectral line but spreads out over frequencies both above and below the nominal signal frequency in the form of modulation sidebands.

One of the most important features that characterize the VCO performance, as well as the price, is the phase noise. Phase noise is a frequency-domain view of the noise spectrum around the oscillator signal. The level of phase noise is a measure of the degree to which an oscillator maintains the same value of frequency over a given time.

In this research, we want to determine the feasibility to exploit the differences between the phase noise characteristics among a set of NICs. In light of this, we provide in this section a description of phase noise and its theory, we explain the model we designed and used for our research, and we explain the measurements we want to obtain. For further information refer to specialized works like [Packard, 1998, Mini-Circuits, , Agilent Technologies, 2005].

Below are three of the most popular ways in which phase noise is defined:

1. The term most widely used to describe the characteristic randomness of frequency stability.

2. The short-term frequency instability of an oscillator in the frequency domain.

3. The peak carrier signal to the noise at a specific offset off the carrier expressed

in dB below the carrier in a 1 Hz bandwidth (dBc/Hz).

3.1 Introduction

A frequency synthesizer is an electronic system for generating any range of frequencies from a single fixed timebase or oscillator. They are found in many modern devices, including radio receivers, mobile phones, radiotelephones, satellite receivers, GPS systems, and in many other devices. In particular, we find them in every 802.11 wireless cards.

Within a frequency synthesizer, the performance of VCO is of paramount importance because it determines many of the overall performance characteristics of the overall system. Oscillators exhibit a natural tendency to amplify any noise present near the oscillation frequency. The closer the frequency of the noise is to the oscillation frequency, the greater the amplification. Noise amplified by the oscillator in this manner is referred to as phase noise. Phase noise is extremely important because it is the most significant source of noise in oscillators. In addition, since the phase noise is centered about the oscillation frequency, it can never completely be removed by filtering.

Phase noise can be understood by recognizing that the phase of an oscillator is arbitrary simply because there is no drive signal present to lock to. Any waveform that is a solution to an oscillator can be shifted in time and still be a solution. If a perturbation causes the phase to be disturbed, there is nothing that acts to restore the phase, so it is free to drift without bound. If the perturbation is random noise, the drift takes the form of a random walk. Furthermore, the closer the frequency of perturbation is to the oscillation frequency, the better it couples to the phase and the greater the magnitude of the drift.

3.2 Noise process in an oscillator

An oscillator can be modeled as a feedback system. To better understand this phenomenon, consider the feedback oscillator shown in Fig. 3.1.

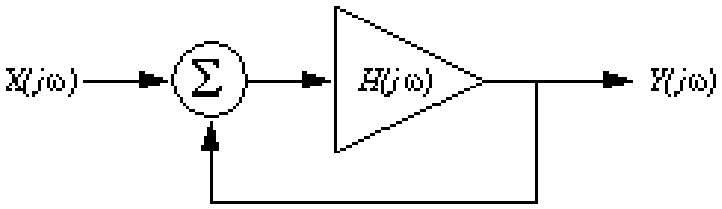


Figure 3.1. General oscillatory feedback system

The loop gain of the oscillator is $H(j\omega)$. $X(j\omega)$ is taken to represent some perturbation stimulus and $Y(j\omega)$ is the response of the oscillator to X. The Barkhausen condition for oscillation states that the effective loop gain equals unity and the phase equals 360 degrees at the oscillation frequency ω_0 (otherwise called carrier frequency) [Clarke & Hess, 1971]. The gain from the perturbation stimulus to the output is

$$\frac{Y(j\omega)}{X(j\omega)} = \frac{H(j\omega)}{(1 - H(j\omega))}\,, \tag{3.1}$$

which goes to infinity at the oscillation frequency f_0, or equivalently $\omega_0 = 2\pi f_0$. When the perturbation stimulus is a noise source, the result is phase noise.

The amplification near the oscillation frequency that generates the phase noise is quantified by assuming the loop gain varies smoothly as a function of frequency in this region [Razavi, 1996].

$$\text{If} \quad \omega = \omega_0 + \Delta\omega \tag{3.2}$$

$$\text{then} \quad H(j\omega) \approx H(j\omega_0) + \frac{dH}{d\omega}\Delta\omega\,, \tag{3.3}$$

and the transfer function becomes:

$$\frac{Y(j(\omega_0+\Delta\omega))}{X(j(\omega_0+\Delta\omega))} \approx \frac{(H(j\omega_0)+\frac{dH}{d\omega}\Delta\omega)}{(1-H(j\omega_0)-\frac{dH}{d\omega}\Delta\omega)}. \tag{3.4}$$

Since $H(j\omega_0)=1$ and $\frac{dH}{d\omega}\Delta\omega \ll 1$ in most practical situations, the transfer function reduces to:

$$\frac{Y(j(\omega_0+\Delta\omega))}{X(j(\omega_0+\Delta\omega))} = \frac{-1}{\frac{dH}{d\omega}\Delta\omega}. \tag{3.5}$$

This equation indicates that a noise component at $\omega=\omega_0+\Delta\omega$ is multiplied by $(\Delta\omega\ dH/d\omega)^{-1}$ when it appears at the output of the oscillator. In other words, the noise power spectral density is shaped by

$$\left|\frac{Y(j(\omega_0+\Delta\omega))}{X(j(\omega_0+\Delta\omega))}\right|^2 = \frac{1}{\left|\frac{dH}{d\omega}\right|^2(\Delta\omega)^2}. \tag{3.6}$$

This is illustrated in Fig. 3.2

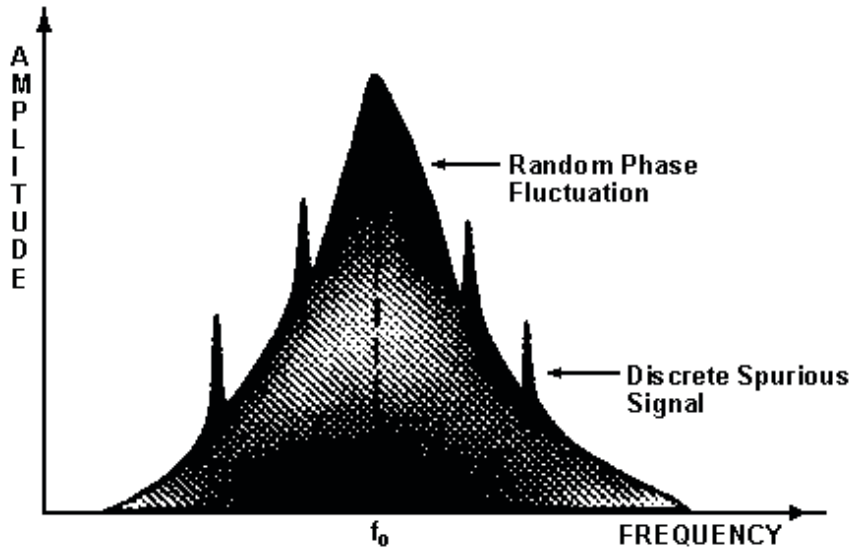


Figure 3.2. Spectrum analyzer display of a sinusoidal signal affected by phase noise

Thus, for circuits that contain only white noise sources, the phase noise voltage (or current) is inversely proportional to $\Delta\omega$ (or equivalently Δf), while the phase noise power spectral density is proportional $1/\Delta\omega^2$ (or $1/\Delta f^2$) near the oscillation frequency.

Phase noise is created by a linear phenomenon, the amplification of noise near the carrier frequency that is a natural consequence of the oscillators complex pole pair on the $j\omega$ axis at ω_0.

In addition, practical oscillators are also strongly nonlinear because they operate in compression. Recall that the loop gain of the oscillator must equal 1 at the oscillation frequency. If the loop gain is less than 1, the oscillation eventually dies out. To ensure the oscillator reliably oscillates even with normal variations in component values, the initial loop gain is always designed to be larger than 1, with the understanding that as the amplitude of the oscillation builds the amplifier goes into compression, which reduces the loop gain. The amplitude stabilizes at the point where the effective loop gain is 1. The nonlinear behavior inherent in practical oscillators results in noise folding. Noise mixes with the oscillation signal and its harmonics get converted up or down in frequency by one or more multiples of the fundamental frequency.

The resulting phase noise plot for an actual oscillator is as shown in Fig. 3.6. The frequency domain response of a source would include terms like Random Walk, Flicker and White Phase Noise to describe the slope of spectral density for given offsets.

3.3 Phase noise and spurious tone

The ideal synthesizer produces a pure sinusoidal waveform

$$V(t) = V_0 \sin(2\pi f_0 t) \tag{3.7}$$

When amplitude and phase fluctuations are accounted, the waveform becomes

$$V(t) = (V_0 + v(t)) \sin(2\pi f_0 t + \varphi(t)) \tag{3.8}$$

where $v(t)$ and $\varphi(t)$ represent amplitude and phase fluctuations, respectively. Because amplitude fluctuations can be removed or greatly alleviated by a limiter or an automatic amplitude control (AAC) circuit [Zanchi et al., 2001b, Zanchi et al., 2001a], we concentrate on phase fluctuation effects in a frequency synthesizer output only. We consider two types of phase fluctuations, the periodic variation and the random variation [Lin, 2000]. In mathematical form, can be written as:

$$\varphi(t) = \Delta\varphi \sin(2\pi f_s t) + \phi(t) \tag{3.9}$$

The first term represents the periodic phase variation, and it produces a spurious tone at an offset frequency of $f_s = f - f_0$ from the carrier frequency f_0. The magnitude of the spurious tone can be derived as follows:

$$\begin{aligned} V(t) &= (V_0 + v(t)) \sin(2\pi f_0 t + \Delta\varphi \sin(2\pi f_s t)) \\ &= V_0 \big[\sin(2\pi f_0 t) \cos(\Delta\varphi \sin(2\pi f_s t)) + \cos(2\pi f_0 t) \sin(\Delta\varphi \sin(2\pi f_s t)) \big] \end{aligned} \tag{3.10}$$

For very small phase modulation, i.e., $\Delta\phi \ll \pi/2$

$$\cos(\Delta\varphi \sin(2\pi f_s t)) \approx 1 \tag{3.11}$$

$$\sin(\Delta\varphi \sin(2\pi f_s t)) \approx \Delta\varphi \sin(2\pi f_s t) \tag{3.12}$$

Then 3.10 yields

$$\begin{aligned} V(t) &\approx \big[\sin(2\pi f_0 t) + \Delta\varphi \cos(2\pi f_0 t) \sin(2\pi f_s t) \big] \\ &= V_0 \big[\sin(2\pi f_0 t) - \frac{\Delta\varphi}{2} \sin(2\pi (f_0 - f_s) t) + \frac{\Delta\varphi}{2} \sin(2\pi (f_0 + f_s) t) \big] \end{aligned} \tag{3.13}$$

From (3.13) we observe that the two spurious tones at $f_0 + f_s$ and $f_0 - f_s$ are both $-20 \log(\Delta\varphi/2)$ dB below the carrier.

The second term $\phi(t)$ in (3.9) represents the random phase variations around the theoretical phase of the signal $\varphi_0 = 2\pi f_0 t$, as illustrated in Fig. 3.3, and produces phase noise. ϕ_{rms} represents the root-mean-square (rms) value of phase noise and it can also be called "mean phase fluctuation".

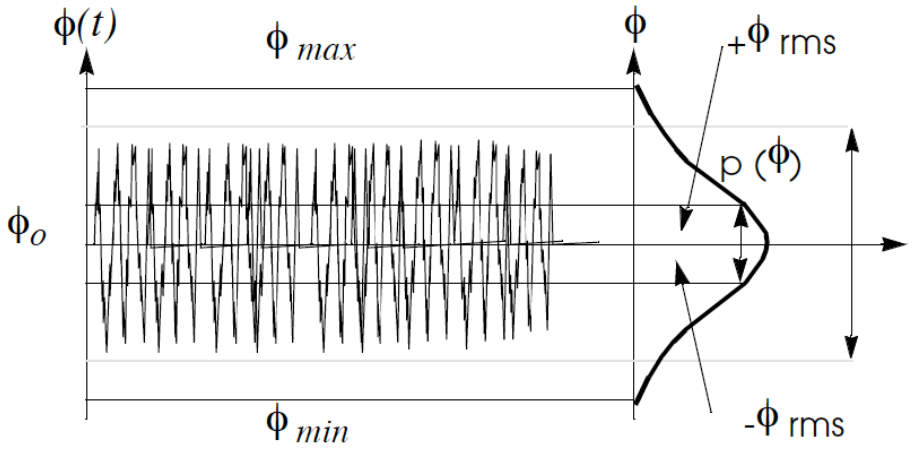


Figure 3.3. Phase fluctuations around the theoretical phase of the signal φ_0

The spectral density of phase variation, expressed in $(\text{radians})^2/\text{Hz}$, is

$$S_\phi(f) = \int_{-\infty}^{\infty} R_\phi(\tau) e^{(-j2\pi f\tau)} d\tau \tag{3.14}$$

where $R_\phi(\tau)$ is the auto-correlation of the random phase variation $\phi(t)$:

$$R_\phi(\tau) = E[\phi(t)\phi(t-\tau)] \tag{3.15}$$

When the rms value of ϕ_{rms} is much smaller than 1 radian, the power spectrum density of $V(t)$ can be approximated as

$$S_V(f) = \frac{V_0^2}{2}[\delta(f-f_0) + S_\phi(f-f_0)] \quad , \tag{3.16}$$

it consists of the carrier power at f_0 and the phase noise power at frequency offsets from f_0. Indeed, the spectrum of an ideal sinusoidal signal (3.7) corresponds to a

DIRAC function at frequency f_0 The disturbances due to the frequency and amplitude fluctuations give rise to a spectral bandwidth and noise sidebands, Figure 3.4.

The single-sideband (SSB) phase noise $\pounds(f_m)$ is an indirect measure of noise energy easily related to the RF power spectrum observed on a spectrum analyzer. $\pounds(f_m)$ is defined as a the ratio of noise power (P_{noise}) in 1 Hz bandwidth at a certain frequency offset $f_m = f - f_0$ from the carrier to the carrier power ($P_{carrier}$) and the unit is dBc/Hz. $\pounds(f_m)$ is usually presented logarithmically as a plot of phase modulation sidebands in the frequency domain, expressed in dB relative to the carrier per Hertz of bandwidth [dBc/Hz].

$$\pounds(f_m) = 10 \log \frac{P_{noise}(1 \text{ Hz at } f_m)}{P_{carrier}} \quad \left[\frac{\text{dBc}}{\text{Hz}}\right] \tag{3.17}$$

Still, when the phase fluctuation peak ϕ_{peak} is $\ll$ 1 radian, it is possible to derive $\pounds(f_m)$ from $S_\phi(f_m)$ using phase modulation theory, as it is described in [Packard, 1998]

$$\pounds(f_m) = 10 \log \frac{S_\phi(f_m)}{2} \quad \left[\frac{\text{dBc}}{\text{Hz}}\right] \tag{3.18}$$

Therefore, the phase noise value observed at a certain frequency offset f_m on the spectrum analyzer is numerically equivalent to $10 \log \lfloor S_\phi(f_m) \rfloor - 3$. For example, Fig. 3.4 illustrates the phase noise and spurs of a synthesized signal of frequency f_0. In this case, the spur level at an offset frequency of $-\Delta f_1$ is 70 dBc/Hz, and the phase noise at an offset frequency of $-\Delta f_2$ is 100 dBc/Hz.

3.4 Measurements

The distribution of these noise sidebands as a function of offset frequency can be expressed in different ways Fig. 3.5.

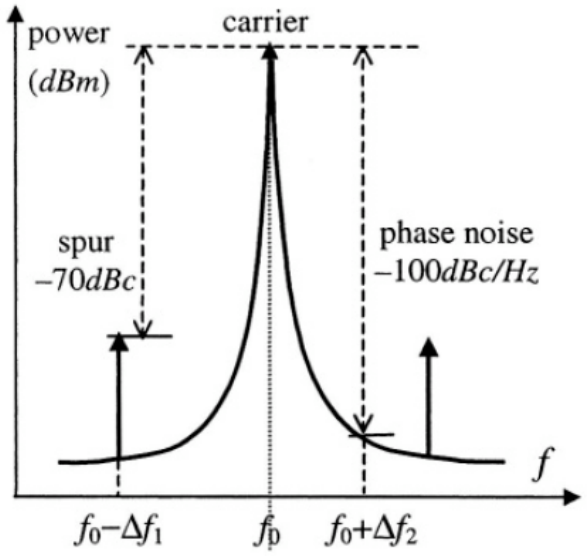


Figure 3.4. Phase noise and spur

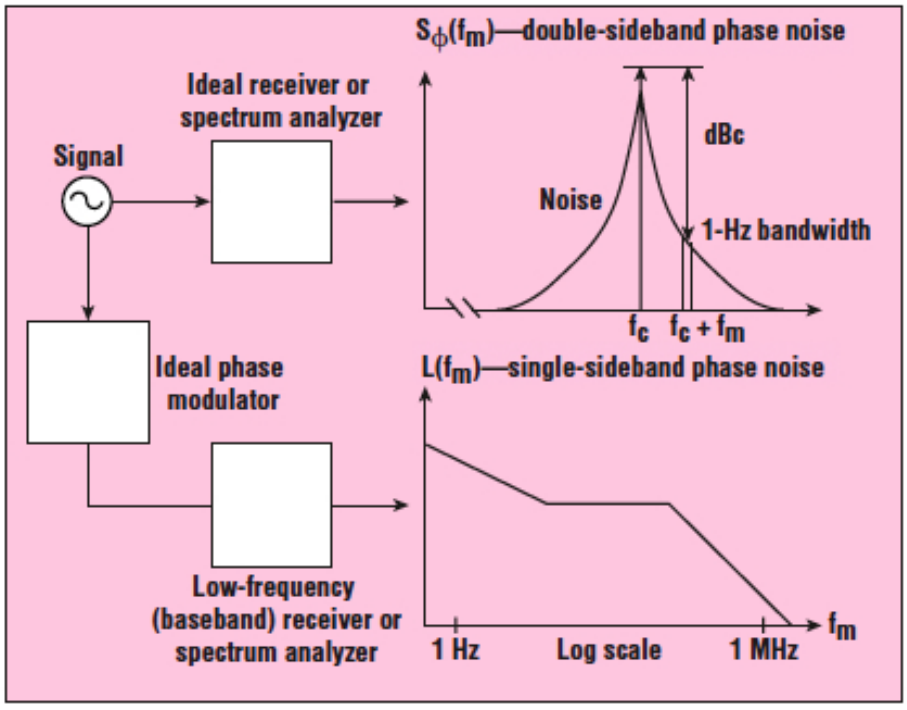


Figure 3.5. A spectrum analyzer can be used to evaluate SSB or DSB phase noise

One way is to use an ideal receiver or spectrum analyzer at RF with a 1-Hz resolution-bandwidth filter. The total power of the signal would first be measured and, since the noise is small, this is essentially equal to the carrier power. Then the receiver would be tuned to a particular offset (f_m) from the carrier, and the phase noise power is measured. The ratio of these two measurements, expressed in decibels, is the normalized power-spectral density (PSD) in one sideband referred to the carrier at a frequency offset f_m. This is known as the single-sideband (SSB) phase noise, $\mathcal{L}(f_m)$, with units of $(\text{Hz})^{-1}$ or expressed in decibels relative to the carrier level as dBc/Hz. A plot of the phase noise as function of the offset is commonly shown in data sheets in order to characterize oscillators and frequency synthesizers.

Another method is to demodulate the signal with an ideal phase demodulator. The output of the phase demodulator is the baseband phase noise and can be analyzed with a low-frequency spectrum analyzer, again with a 1-Hz resolution-bandwidth filter. The resulting plot as a function of baseband or offset frequency is the double-sided phase-noise spectrum, $S_\phi(f_m)$ $[(\text{radians})^2/\text{Hz}]$. As we have seen in Eq. (3.18), the double-sideband (DSB) phase noise is twice that of (or 3 dB more than) the SSB phase noise.

Noise has infinite bandwidth, and hence the greater the bandwidth of the instrument being used to measure a carrier frequency with noise, the higher the noise it measures. For example, as you change the resolution bandwidth (res. BW) (equivalent to the physical bandwidth of the IF channel) on a spectrum analyzer, the noise magnitude changes. The industry has settled the standard on a correlation bandwidth for phase noise measurements of 1 Hz, known as the normalized frequency. There are few spectrum analyzers that have a 1 Hz resolution bandwidth and such a spectrum analyzer is very expensive. In fact, the closer to the carrier you want to measure, the higher the instrument cost will be. A spectrum analyzer will specify

how close to the carrier it can measure (known as the lowest resolution bandwidth possible); above this maximum frequency, one can normalize the reading to 1 Hz with the following:

$$\underbrace{\mathcal{L}(f_m)}_{\text{1-Hz BW}} = \underbrace{\overline{\mathcal{L}}(f_m)}_{\text{un-normalized BW}} - 10\log\,(\text{res. BW of spectrum analyzer}) \tag{3.19}$$

3.5 Simulation model

3.5.1 General case. The phase noise $\varphi(t)$ consists of deterministic components and random noise. For example, temperature change, supply voltage, and the output impedance of the oscillator are included among deterministic components. Ignoring the deterministic effects with the exception of the frequency drift, a general PSD model of the random phase variation $\phi(t)$ comprises five terms:

$$S_\phi(f) = \underbrace{k_0}_{\text{(i)}} + \underbrace{\frac{k_{-1}}{f}}_{\text{(ii)}} + \underbrace{\frac{k_{-2}}{f^2}}_{\text{(iii)}} + \underbrace{\frac{k_{-3}}{f^3}}_{\text{(iv)}} + \underbrace{\frac{k_{-4}}{f^4}}_{\text{(v)}} \tag{3.20}$$

where: (i) is white phase noise, (ii) is flicker phase noise, (iii) is random phase walk or white frequency noise, (iv) is flicker frequency noise and (v) is random frequency walk.

When the phase noise is plotted in decibels versus log frequency, various regions of the phase-noise curve can be identified. These regions have slopes of 0, $1/f$ (-10 dB/decade), $1/f^2$ (-20 dB/decade), $1/f^3$ (-30 dB/decade), etc. Fig. 3.6.

The flat or zero-slope region corresponds to white phase noise of thermal origin. This thermal, or Johnson, noise has a Gaussian amplitude distribution, constant with frequency. Amplifier noise figure is a manifestation of this thermal noise. Close to "DC" or a zero-frequency offset, there is a region of $1/f$, or flicker noise. This is believed to come from irregularities in the semiconductor structure. Frequency

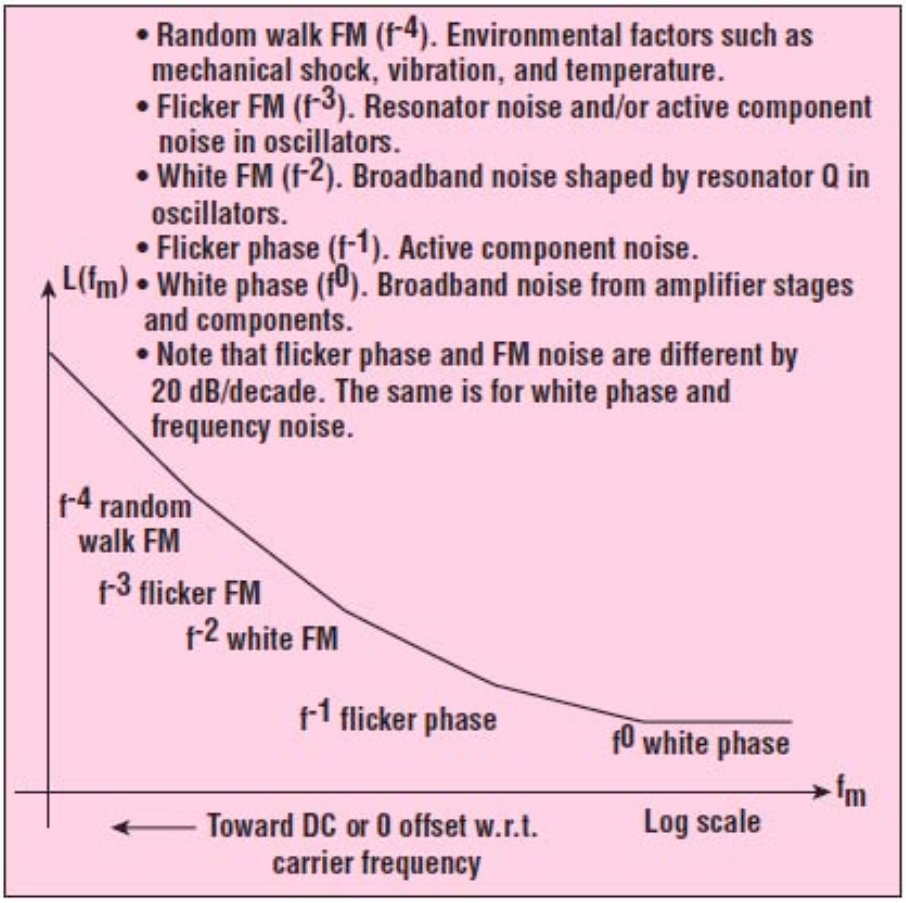


Figure 3.6. Phase noise, in the most general form, consists of several components, including random-walk FM, flicker-noise FM, white-noise FM, flicker phase noise, and white phase noise

dividers and amplifiers exhibit only 0 and $1/f$ slope regions. Oscillators can have all regions.

3.5.2 Our model. As we have seen, the most general form to express mathematically phase noise is expressed by Eq. (3.20) and, when it is plotted in a graph decibels versus log frequency, it can be composed of several regions, as we have seen in Fig. 3.6.

We remember that our goal is to demonstrate the feasibility to exploit phase noise in a WIS. Since all the equipments such as wireless network cards, spectrum analyzer and a laboratory to conduct experiments to be available for several months can be very expensive, we have decided to solve the problem at first through computer simulations and face a real experiment, as a future work, only if the results of our simulations are satisfactory.

For all these reasons, our goal is to simulate the complete system and their RF impairments, included the phase noise, with a sufficient level of details to permit us to understand if this approach is feasible or not, but without the pretension to simulate every component in a perfect manner.

We have decided to use Matlab/Simulink as the simulation environment; here we describe the Simulink built-in phase noise block [Mathworks,], shown in Fig. 3.7, based on a work of Kasdin [Kasdin, 1995]. Since this paper is quite complicated, we suggest to refer to the original work [Kasdin, 1995] for further details of the implementation. Here we provide a description of the block and the effects on digital modulated signals based also on [Mathworks,].

The phase noise block, in Fig. 3.7, applies phase noise to a complex, baseband signal. This block applies the phase noise as follows:

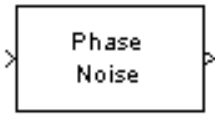


Figure 3.7. Phase noise block

1. Generates additive white Gaussian noise (AWGN) and filters it using a digital filter.

2. Adds the resulting noise to the angle component of the input signal.

The phase noise block generates phase noise over the entire spectral observation window, from 0 Hz (or as close as possible to 0 Hz) to $\pm\frac{F_s}{2}$, where F_s represents the sampling frequency. The noise is scaled so that it is at the block-specific phase noise level at the specified frequency offset. The block generates a phase noise with characteristic over the entire frequency range.

The block's implementation of phase noise is shown in Figures 3.8 and 3.9:

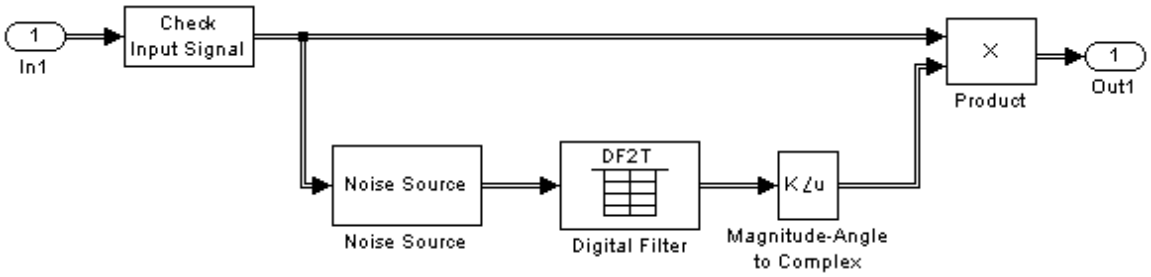


Figure 3.8. Phase noise subsystem

The effects of changing the block's parameters are illustrated by the following scatter plots of a signal modulated by 16-ary quadrature amplitude modulation (QAM). The usual 16-ary QAM constellation without distortion is shown in Fig. 3.10:

Figure 3.11 shows a scatter plot of the same signal, this time using the Phase

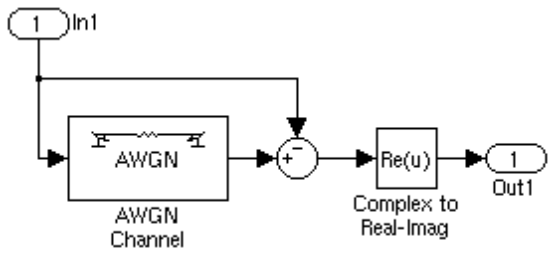


Figure 3.9. Noise source subsystem

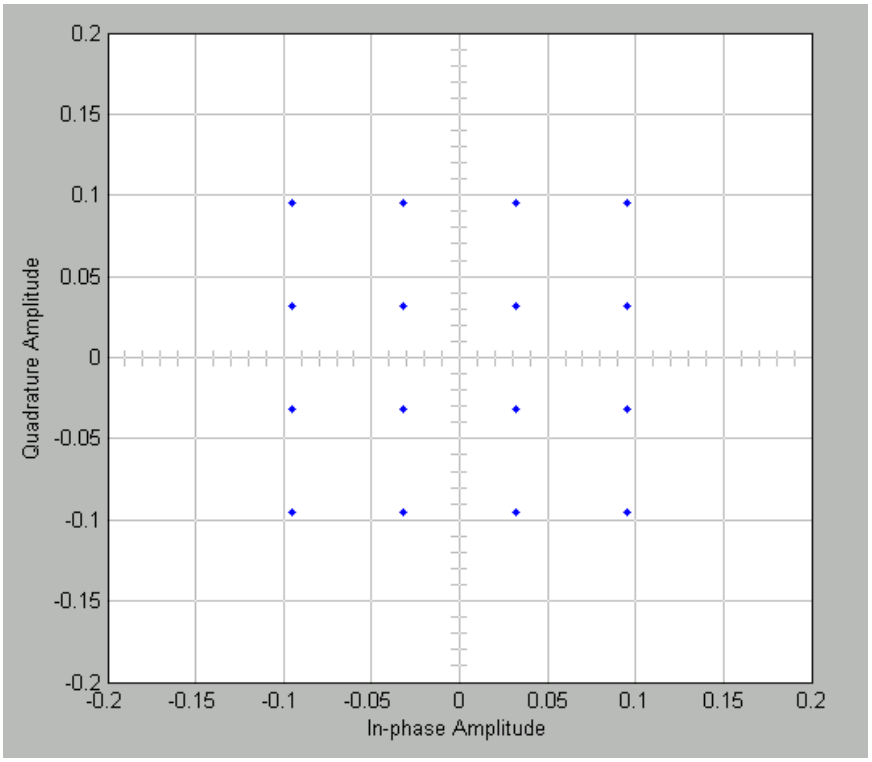


Figure 3.10. Scatter plot of an ideal 16-QAM signal

Noise block with Phase noise level (dBc/Hz) set to -70 and frequency offset set to 100 Hz:

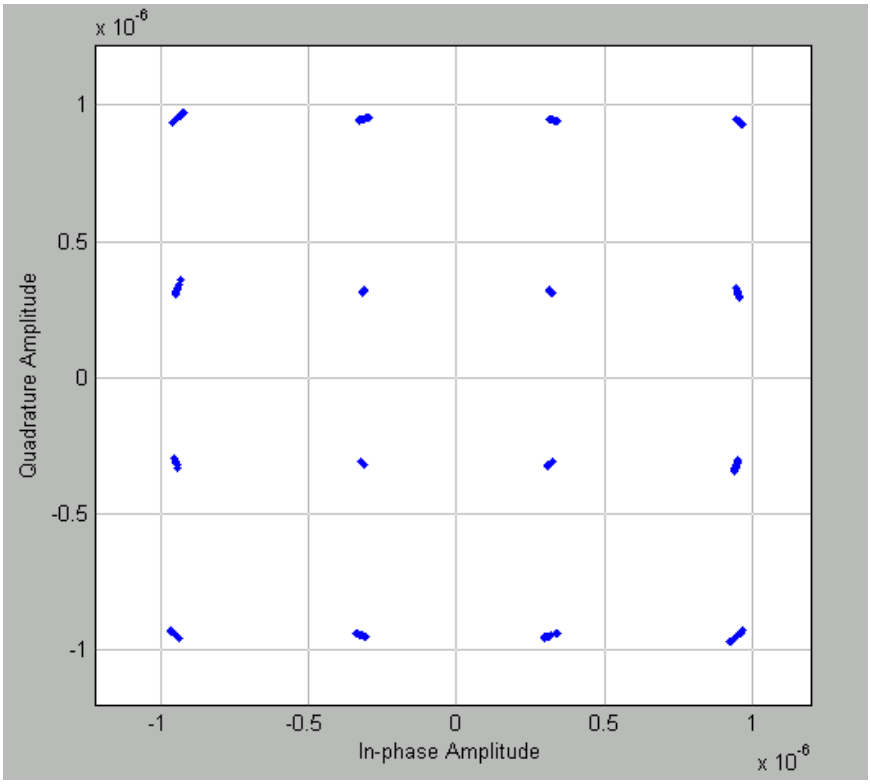


Figure 3.11. Scatter plot of an ideal 16-QAM signal with phase noise

In terms of SSB, the phase noise characteristic is shown in Fig. 3.12

It is clear how our phase noise model characteristic is a simplified version of a real one. It is possible to graphically understand the different comparing Fig. 3.6 and Fig. 3.12. Our model includes only the flicker phase noise (see Eq. (3.20)), which is found in all components, and is characterized by its spectral density variation slope $1/f$

Although our model does not take into account all the possible slopes an oscillator can have in its SSB characteristic, for our purpose it is sufficient. Indeed, in commercial NIC data sheets [Skyworks, , Crystek Corporation, , DrawCom Pty Ltd,], the phase noise is usually specified with a number of sampled values, usually between

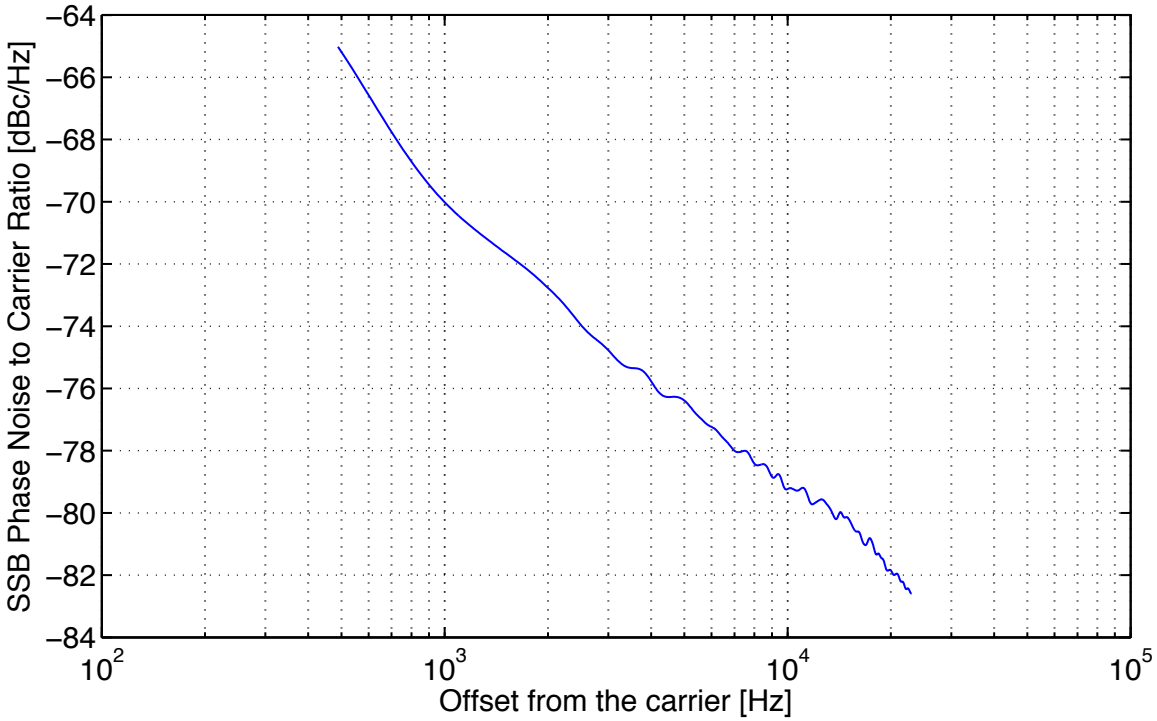


Figure 3.12. Single-sideband characteristic of phase noise

one and three, at specific frequency offsets from the carrier.

Table 3.1. Example of phase noise values for a commercial IEEE 802.11 data sheet [DrawCom Pty Ltd,]

Offest	dBc/Hz
1 kHz	72
10 kHz	80
100 kHz	95

Since our model is able to simulate only $1/f$ or flicker phase noise, we are able to simulate only one slope of the phase noise SSB characteristic; because of which, we will measure the phase noise only at one specific offset from the carrier. A better simulation model would allow us to simulate different slopes and, furthermore, to measure the phase noise at different offsets; this fact would able us to capture more information from phase noise measurements and to improve our results based

on phase noise analysis.

As we have already stated, a very realistic phase noise model is not the main goal of this research, and we will base our analysis on the simplified phase noise model described in this version. We think this level of detail is sufficient to demonstrate the feasibility (or the unfeasibility) of this approach.

As we will see in the next chapters, our research includes the implementation of an existing approach that we have decided to improve. An indirect validation of our model comes from the comparison of the performance of this approach obtained by the authors in laboratory, and our computer simulation. They behave in a similar way in all the tests the authors did and that we replayed.

CHAPTER 4

WIRELESS IDENTIFICATION FROM A PHASE NOISE PROSPECTIVE

At the moment we started our research, instead of creating a completely new wireless identification system, we preferred to improve an existing one, trying to combine all the advantages of our idea and the chosen approach.

Although transient-based approaches are the most popular in literature for physical-layer identification systems, we decided to not consider them because:

1. Transient analysis offers good classification performance only where the beginning and end of the transient can be reliably identified. The nature of transients is such that they are difficult to detect and there is no obvious correct way to succinctly describe them. The latter property is reflected in the amount of literature of the subject [Shaw & Kinsner, 1989], [Ureten & Serinken, 2007], [Hall et al., 2005], [Hall, 2006], [Rasmussen, 2007], [Ureten & Serinken, 1999], [Hall et al., 2003], [Barbeau et al., 2006].

2. It has been reported in [Ellis & Serinken, 2001, Gerdes et al., 2006], that this approach is not always able to distinguish between same manufacturer/same model variants.

3. It needs specific hardware

Very recently, Brik et al. [Brik et al., 2008] proposed a device identification based on the variance of modulation errors. The authors explore the possibility to perform wireless identification based on information extracted from the transmitted signal in the modulation domain.

We chose to implement and improve PARADIS because:

1. We found PARADIS's approach both extremely smart and easily implementable because, operating in the modulation domain, it takes advantage on the strict defined structure of signals imposed by the modulation scheme. Because of that, the comparison between received symbols and ideal ones could not be easier.

2. Physical modulations schemes are designed explicitly to protect encoded data against adverse channel conditions. Therefore, signal representation using symbols in modulation domain is more stable and resilient to effects of noise that distorts raw waveforms. Because its input is more stable, PARADIS performs well in the presence of noise. It also avoids all the problems and concerns related to the transient-based approaches.

3. The technique is completely passive. No NIC is required to communicate with the WLAN infrastructure and, within the 802.11 wireless range, it is not possible to hide or ovoid this kind of identification. Therefore, PARADIS is completely transparent to users and does not require any modifications to client-side hardware and software.

4. Our idea, i.e. exploiting the phase noise to perform identification, is easily implementable in PARADIS.

5. PARADIS is one of the most promising wireless identification system in terms of identification rates.

Since PARADIS is the starting framework we have used to accomplish our research, we will first explain in details how it works in Section 4.2.

4.1 Transmitter individuality

Wireless identification is possible because due to normal variations and imperfections in physical properties of the components that compose wireless devices.

In general, these imperfections are called RF impairments because they cause the device's emissions to be different from the theoretically ideal output.

These impairments have been deeply measured and analyzed in communication theory [Agilent Technologies, c], [Agilent Technologies, a], [Agilent Technologies, b], [Agilent Technologies, 2001], [Stepanek & Kilpatrick,] because they heavily affect the performance of a digital transmission scheme. Even when constructed using the same manufacturing and packaging processes, no two NICs are perfectly identical.

Although it might be possible to eliminate these hardware differences through more precise manufacturing and quality control, it can also greatly increase costs. Manufacturers allow such impairments in their devices because, in general, communication designers have taken into account them. In fact, a wireless network card, even with such minor impairments, continues to work well within the tolerances specified by the IEEE 802.11 communication standard. This work wants to extend the concept of *radiometric identity*, i.e. the unified effect of its impairments in the modulation domain, proposed and demonstrated by Brik et al. [Brik et al., 2008], that can be used to discern between different 802.11 NICs.

4.2 Passive Radiometric Device Identification System

PARADIS, otherwise called Passive RAdiometric Device Identification System [Brik et al., 2008], stands out from other radiometric identification approaches because it leverages understanding of 802.11 PHY layer to reduce the complexity of the underlying problem.

The brilliant idea behind the PARADIS is to exploit well defined metrics in the constellation domain instead of a compact representation of the transient.

As we have seen along this section, several research groups have worked to find a suitable and compact representation for the transient. Unfortunately, since

the factors that determine a transient's shape are poorly understood, and it lasts under $2\mu s$ for 802.11 transmission [IEEE Standards Ass., 1999], the transient is not used in any protocol functions. Therefore, transients have to be treated essentially as arbitrary waveforms, and finding a compact representation that will be effective for all possible transients involves heuristics and guesswork.

In contrast, modulation gives a waveform a well-defined structure of limited complexity, making operations on it intuitive to understand and straightforward to implement. Indeed, minor variations in analog hardware of transmitters are manifested as impairments not only in the transient of the transmitted signal, but also in all the symbols that compose a 802.11 frame.

PARADIS processes frame-by-frame and extracts some features that are affected by hardware RF impairments, such as frequency, magnitude/phase errors, I/Q origin offset and SYNC correlation comparing the received frame to the ideal one in the modulation domain. After a normalization, these features are gathered in a real vector that represents the device's signature linked to that frame. Note that although experimental section of this work deals exclusively with IEEE 802.11, PARADIS is more general and can work with any communication standard that uses digital modulation. This is because the metrics listed above are defined for any digital communication standard.

4.2.1 IEEE 802.11 background. The physical layer of the IEEE 802.11 standards use different I/Q digital modulation techniques depend on the amendment and the signal quality. As the name I/Q signifies, data is encoded using two independent carrier components, or sub-carriers. These sub-carriers are called in-phase (I) and quadrature (Q), because they are separated in phase by $\pi/2$. Symbols of an I/Q modulation scheme are defined using a constellation diagram where different symbols are represented as points in I/Q space, or modulation domain. Depending on modu-

lation scheme, a single symbol can encode multiple data bits. For example, in QPSK modulation each symbol encodes two data bits, as shown in 4.1.

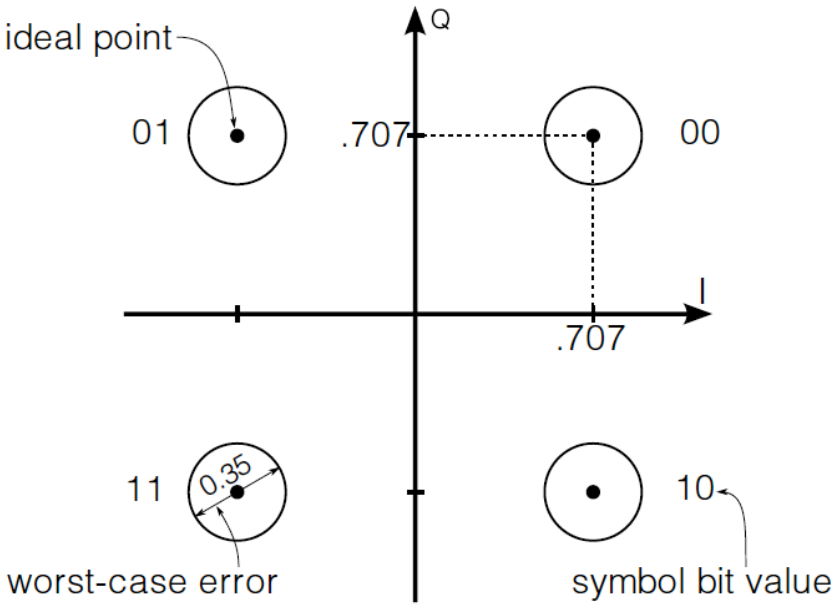


Figure 4.1. The 4 symbols of QPSK on I/Q plane

To give another example, consider sending a bit sequence "1000" using QPSK. First, the transmitter modulates the carrier wave to correspond to I/Q value of (0.707, -0.707) to send the first two bits (10), and then transitions the carrier to (0.707, 0.707) to send the next two bits (00).

4.2.2 Errors in modulation domain. As we have already said, PARADIS exploits errors in the modulation domain to perform identification. Modulation errors can be caused by hardware impairments (what we want to measure), channel characteristics and noise at the receiver. Modulation errors are typically measured by comparing phasors, or vectors corresponding to the in-phase and quadrature values of a signal at the instant in time when a symbol is detected. In our context, phasors could be thought as vector representations of symbols.

In this section we want to clarify the kind of errors that can be measured.

Relevant error metrics are explained below and illustrated in Fig. 4.2:

- *Phase error*: the angle between the ideal and measured phasor.
- *Magnitude error*: the difference in magnitudes of the ideal and measured phasor.
- *Error vector magnitude (EVM)*: the magnitude of the vector difference between the ideal and measured phasor.

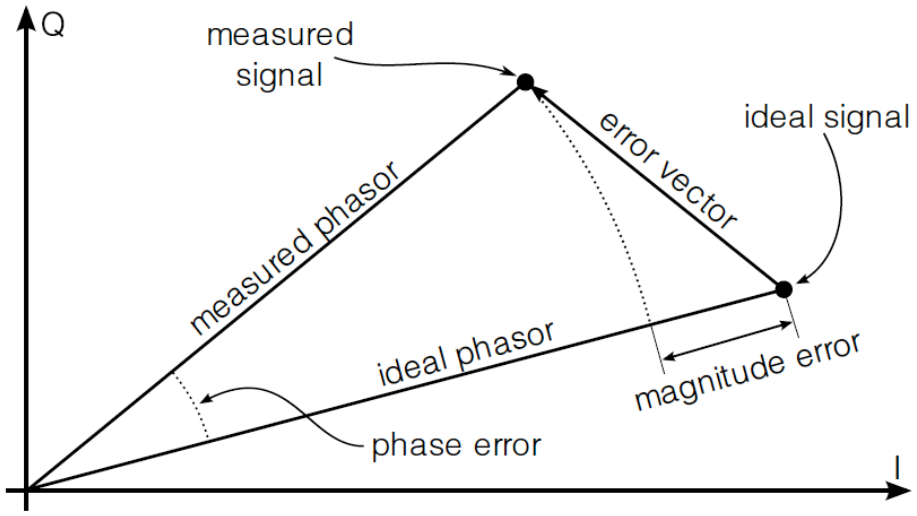


Figure 4.2. Modulation errors

Since a single symbol in a frame represents a very small amount of information, a classification algorithm is not able to take a decision based only on that. Therefore, the authors have used the above terms to describe the average errors across all symbols in a frame, rather than a specific symbol. In this way, modulation errors linked to a dispersive channel, fading or noise, are mitigated and the impairment errors are pointed up.

In contrast, the following error metrics are only defined for an entire frame:

- *I/Q origin offset*: the distance between the origin of the ideal I/Q plane and the origin of the observed symbols.
- *Frequency error*: the difference between the ideal and observed carrier frequency. This is the amount by which the receiver's frequency had to be adjusted from channel center to achieve carrier lock.
- *SYNC correlation*: the correlation of I/Q values from an observed and the ideal SYNC, which is a short signal that precedes encoded data and is used to synchronize the transmitter and the receiver.

Table 4.1. Modulation error metrics for IEEE 802.11a, channel 36, 2Mbps, QPSK

Error type	unit	reference	range	definition
frequency	Hz	5.180 GHz	± 207.2	± 20 ppm f_c
phase	$^\circ$	ideal symb	± 10	$\arcsin(E_{max})$
magnitude	n/a	ideal symb	± 0.17	$\pm E_{max}$
EVM	n/a	ideal symb	[0, 0.35]	$\leq 2E_{max}$
I/Q offset	n/a	ideal origin	[0, 0.17]	$\leq E_{max}$
SYNC	%	max corr.	[0, 1]	correlation
f_c = channel frequency E_{max} = max I/Q error				

The IEEE 802.11a standard [IEEE Standards Ass.,] specifies error tolerance for these metrics with respect to the ideal signal. In Table 4.1 we summarize them. For example, in QPSK modulation, the symbol error vector magnitude tolerance is 0.35. Similarly, the frame frequency error tolerance for the IEEE 802.11a standards is $\pm$ 20 ppm (parts per million). Hence, for 802.11a channel 36 centered at 5.180 GHz, valid frames need to have center frequency in the 207.2 kHz band around the channel center. Note that, although error tolerances may be specific to IEEE 802.11, the error metric we defined are general and can be used with any standard that uses I/Q modulation, not just 802.11.

Not surprisingly, we observed that classification accuracy can be improved if it is performed on multiple frames, rather than just one. Noise and interference, present in the channel, have a random structure. On the other hand, RF impairments affect the signal and, moreover, the metrics in a more fixed way. Averaging on more frames, we are going to mitigate random variation effects and amplify fixed distortions due to RF impairments.

4.3 Phase noise analysis

Our approach, that we call phase noise analysis (PN-analysis), uses as framework the PARADIS system proposed by Brik et al. [Brik et al., 2008] very recently. This is a WIS that operates in the modulation domain. It defines several metrics, related to modulation errors, well-defined in every communication standard, for example the one we are focused is IEEE 802.11 and a summary of these metrics is shown in Table 4.1. We implemented our idea acting on the signature generator and classifier proposed by Brik et al. We defined a new metric based on phase noise measurements at the receiver, because it is a feature that uniquely characterize every transceiver. Our main goal is to demonstrate its usability in wireless identification.

We will explain the details of our model and the measurement method in Section 4.5, but we want to explain here the basic idea. All the symbols the transmitter sends via wireless transmission are affected by several RF impairments. The PARADIS approach we used as framework already studied the effect of these distortion on the modulation metrics defined by the standard.

Our contribution is the definition of a new metric based on phase noise, that is an important impairment present in every oscillator. It affects every symbol with a random phase error; the receiver can measure this error computing the difference between the detected symbol, that we assume as the ideal one, and the received

symbol. Gathering this data for all the symbols in a frame, it is possible to calculate the SSB phase noise as explained in Section 3.4.

We want to underline that, since other factors can affected the phase of the symbols, for instance noise, if the receiver makes a mistake in the detection phase, the assumed ideal symbol will be different from the originally transmitted one. It is clear how this kind of detection errors can negatively affect our phase noise estimation and, during the simulation, we will test how our PN-analysis behaves in bad signal quality conditions.

4.4 Metrics

It is important to understand that the performance of a radiometric system cannot be summarized by a single number. It is possible to take different points of view to observe and evaluate the same system, and the results can differ in relationship on what we are looking for. To evaluate these radiometric identification systems we have chosen three metrics. The first one is the most general and will be used to compare PARADIS and our approach. The other two will look deeper to the card-by-card performance.

The metrics are:

Average error rate (AER): AER, otherwise called average misclassification rate, is the ratio of all misclassified samples to the total number of samples in the data-set. We will only use average error rate as a rough measure of overall classifier performance. One of the reasons we do not use this metric more is because it does not differentiate between false positives and false negatives.

False reject rate (FRR): FRR estimates the likelihood of a NIC's frame being incorrectly identified as someone else's. The metric is particularly important because

a false positive, or false alarm, from the point of view of an administrator happens when the WIS rejects identity of a legitimate user, that is, a false negative from the point of view of the WIS. Therefore, in this case false reject rate corresponds to the likelihood of the WIS wasting administrator's time, and must be minimized.

Worst-case similarity (WCS): one of the important performance measures that error rates do not capture is how uniformly the misclassifications are distributed across the population. In particular, it is important that no NIC is consistently misidentified as another one, since otherwise that NIC would be able to masquerade as someone else. To quantify this aspect of performance we use the measure of a NIC's worst-case similarity in the following way. Given a NIC, the victim, we find among the other NICs the one with the greatest fraction of frames misclassified as the victim. This is the worst-case, or most dangerous impostor, and the fraction of its frames misclassified as the victim is the worst-case similarity of the victim. For example, suppose a NIC has worst case similarity of 0.5, it means there is another transmitter, in the population, half of whose frames are misclassified as coming from the NIC.

4.5 Simulation Model

4.5.1 Simulink Model. In this Section we provide a description of our model. In Fig. 4.3 is possible to see the high level model composed of several blocks: (i) transmitter (ii), AWGN channel, (iii) receiver and (iv) BER calculation. Now we will describe the details of every block. We have decided, for simplicity and without loss of generality, to create a baseband model.

Note that although this work deals exclusively with IEEE 802.11, the PARADIS approach we have implemented and our PN-analysis are more general and can work with any communications standard that uses digital modulation. This is because all the metrics used by PARADIS are defined for any digital communication stan-

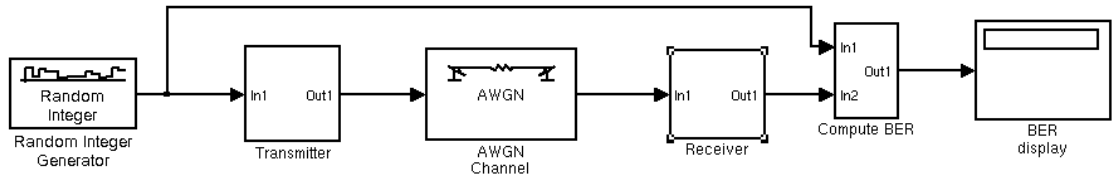


Figure 4.3. Simulink model

dard and it is possible to measure the phase noise level, for example with a spectrum analyzer, for every transceiver.

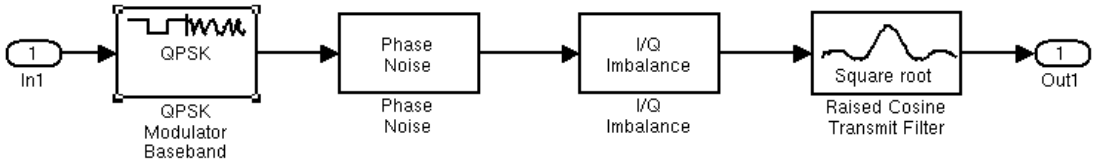


Figure 4.4. Transmitter model

The details of the first block, the transmitter, are shown in Fig. 4.4 The physical layer of the IEEE 802.11 standards uses different I/Q digital modulation techniques which depend on the amendment and the signal quality. In particular, we have focused on one of the most popular amendments, 802.11b, with modulation scheme QPSK, bit rate of 2 Mbps, channel 1 at 2.142 GHz.

Within this choice, the IEEE 802.11b standard [IEEE Standards Ass., 1999] specifies error tolerance for the metrics we are going to measure with respect to the ideal symbols. In Table 4.2 we summarize them for the chosen case. In particular, the symbol error vector magnitude (EVM) tolerance is 0.35, the frame frequency error tolerance is $\pm$ 25 ppm, and valid frames need to have center frequency in the 60.3 kHz band around the channel center.

As it is possible to see in Fig. 4.3, we have implemented an AWGN channel

Table 4.2. Modulation error metrics for IEEE 802.11b, channel 1, 2Mbps, QPSK

Error type	unit	reference	range	definition
frequency	Hz	2.142 GHz	± 60.3	± 25 ppm f_c
phase	$^\circ$	ideal symb	± 10	$\arcsin(E_{max})$
magnitude	n/a	ideal symb	± 0.17	$\pm E_{max}$
EVM	n/a	ideal symb	[0, 0.35]	$\leq 2E_{max}$
I/Q offset	n/a	ideal origin	[0, 0.17]	$\leq E_{max}$
SYNC	%	max corr.	[0, 1]	correlation
f_c = channel frequency $\quad$ E_{max} = max I/Q error				

and we will test our approach with different SNRs. The receiver, shown in Fig. 4.5, is basically the mirror version of the transmitter with some additional blocks for phase noise and modulation errors measurements. Some blocks have, in the caption, the writing "to workspace". It is because we perform some post processing algorithms after the communication has ended to generate, finally, the radiometric signature that uniquely characterize the network card.

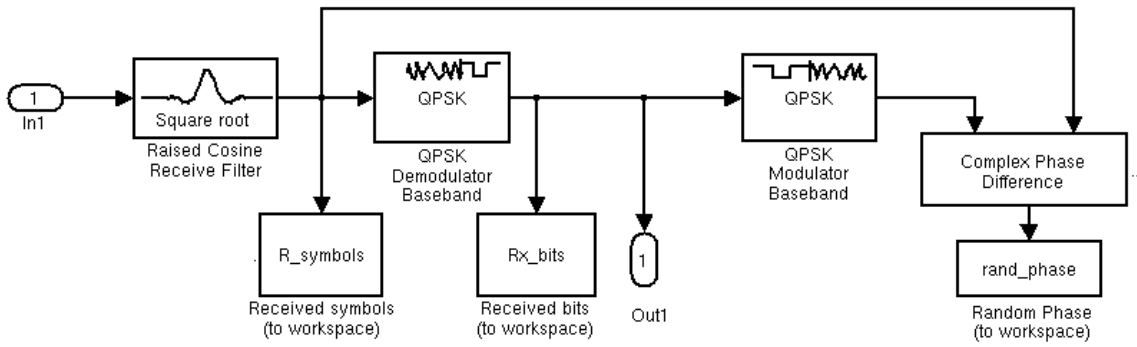


Figure 4.5. Receiver model

In every digital modulation scheme, data is encoded using two independent carrier components, or sub-carriers, called in-phase (I) and quadrature (Q), because they are separated in phase by $\pi/2$. In QPSK modulation, symbols in the constellation diagram are determined by the phase in the sine and cosine waves used to

transmit them:

$$s_n(t) = h_{Tx}(t)\cos(2\pi f_0 t + \varphi_n) \qquad t \in \mathbb{R} \qquad n = 1, 2, 3, 4 \tag{4.1}$$

where $h_{Tx}(t)$ is a real-valued finite energy baseband pulse, and let the phase φ_n be:

$$\varphi_n = \frac{\pi}{4}(2n - 1) \qquad n = 1, 2, 3, 4 \tag{4.2}$$

An alternative expression of Eq. (4.1) is:

$$s_n(t) = Re[e^{j\varphi_n} h_{Tx}(t) e^{j2\pi f_0 t}] \tag{4.3}$$

As we have seen in Fig. 3.10, the phase noise affects the transmitted symbols altering the phase, i.e. creating a random rotation of the symbols in the constellation, as we have shown in Fig. 3.11

Considering only the effect of phase noise, we can write the n-th received symbol $r_n(t)$ as:

$$r_n(t) = Re[e^{j\varphi_n} h_{Tx}(t) e^{j2\pi f_0 t} e^{\phi(t)}] \tag{4.4}$$

where $\phi(t)$ is random phase variation, as defined in Eq. (3.8) and (3.9).

The additional blocks in the receiver perform a complex phase difference between the received symbols $r_n(t)$ affected by phase noise and the detected symbols $\bar{s}_n(t)$. If the signal quality is sufficient for a correct detection, the detected symbol $\bar{s}_n(t)$ and the transmitted one $s_n(t)$ are equivalent. Unfortunately, if the receiver makes a mistake during the detection phase because, for example, the SNR is not sufficient, the detected symbol will differ from the transmitted one affecting negatively the phase noise measurement.

Since the phase noise is essentially a statistical phase error, more the data its estimation based on, better it is. To mitigate wrong detections, we base our PN-

analysis on a multi frames base. We also evaluate how many frames are need for a good measurement.

At the end of the process, phase fluctuations $\phi(t)$ data are saved on the computer and used to calculate its PSD $S_\phi(f)$ using Eq. (3.14). Through Eq. (3.18) it is finally possible to obtain the $\mathcal{L}(f)$ SSB phase noise.

All the post processing functions are implemented as MATLAB scripts. They carry out all the calculations both for PARADIS and PN-analysis for the respective metrics.

4.6 Signatures in practice

So far, we have only described the high level concepts about this approach but we have not clarified in details how a signature is generated, classified and used.

PARADIS works in a frame-by-frame basis. When a WIS node receives a wireless frame, it elaborates all the symbols that composed it and calculates the averaged measurement of the chosen metrics, that are: (i) magnitude, (ii) phase, (iii) I/Q origin offset and (iv) SYNC correlation. For practical reasons, it is advantageous to scale and normalize data before applying a classification algorithm. Therefore, in our implementation a signature of a frame is a real vector whose elements correspond to our metrics, and range in values from 0 to 100. Normalization is performed according to the valid range of a given error metric, which is ultimately defined by the communication standard and the NIC data sheet for the phase noise value. Table 4.2 summarizes valid ranges of modulation accuracy metrics for IEEE 802.11b in the case of bit rate 2 Mbps, QPSK modulation scheme, channel 1. Note that EVM is not used either by PARADIS, or by our approach, and is included for completeness only. PN-analysis shares with PARADIS all these metrics and the elaboration process, implemented as MATLAB scripts. We implemented our idea defining a new metric based on phase

noise measurements. Practically, the difference is that PARADIS is based only on modulation metrics, while our approach also considers phase noise and, moreover, the signatures generated by PN-analysis have one more element. In this way we are immediately able to compare our performance and to evaluate if our approach is feasible or not.

See Figures 4.2 and 4.2 for intuition on derivation of valid ranges for modulation errors, Chapter 3 for phase noise description and Fig. 3.4 for intuition of phase noise measurements.

4.6.1 Classification algorithm. As in PARADIS, we perform identification using a classifier that links a frame signature to a particular NIC. In this thesis we have decided to implement only the k-Nearest-Neighbor (kNN) algorithm [Mitchell, , Wikipedia, e], because of its simplicity and better performance with this problem.

kNN classifier is an instance-based learning algorithm that is based on a distance function for pairs of observations, such as the Euclidean distance or the Manhattan distance. In this classification paradigm, k nearest neighbors of a training data are computed first. Then the similarities of one sample from testing data to the k nearest neighbors are aggregated according to the class of the neighbors, and the testing sample is assigned to the most similar class. A major drawback of the similarity measure used in kNN is that it uses all features equally in computing similarities. This can lead to poor similarity measures and classification errors, when only a small subset of the features is useful for classification.

We want to underline that it is not our goal to find the best-performing algorithm for wireless identification and for the purpose of this work we do not see the need to evaluate either alternative algorithms, or alternative classifier implementations.

From the implementation standpoint, kNN is typical data-agnostic classifier

that takes one or more real vectors corresponding to signatures and identifies corresponding NIC as an integer between 1 and the number of NICs represented in the training set.

Rather than repeating the details of implementation that can be found in references [Mitchell, , Dudani, 1991, Patrick, 1991], we will illustrate how it works with an example. Those familiar with the basics of kNN algorithm may want to skip the corresponding sections below.

4.6.1.1 Training phase. The kNN algorithm needs a training phase, when it records all the signatures of our set of NICs. To do this, the WLAN administrator simply has the NIC transmit frames for a short period of time. Because of this assumption, we never consider the SNR during the training phase as a parameter because the administrator carries out the training phase in optimal conditions. Furthermore, we assume an SNR of 20 dB in this situation.

To decide how many frames we need, we have to test both approaches with different training set size, as it will be shown subsequently. In practice with a training set size of 20 frames we reach the best performance.

For simplicity, suppose that NIC A broadcasts four frames with the following radiometric signatures, as defined in Section 4.6:

$$\begin{aligned} A_1 &= [0.1, 0.9, 0.3, 0.7, 0.6] \qquad A_3 = [0.1, 0.8, 0.1, 0.6, 0.4] \\ A_2 &= [0.2, 0.7, 0.1, 0.6, 0.4] \qquad A_4 = [0.1, 0.9, 0.9, 0.7, 0.6] \end{aligned} \tag{4.5}$$

In order to train, this classifier requires two kinds of functionality: signature manipulation using matrix operations, and a distance function that computes a scalar measure of similarity between two signatures. It uses the l_1 or Manhattan distance [Wikipedia, b], which is the sum of absolute values of component-wise differences between two signatures. During training, it first discards outliers one-by-one until

only half of the training signatures remain. If there are n signatures, an outlier is defined as the signature farthest away from the average:

$$\arg\max_{A_i} l_1\Big(\frac{1}{n}\sum_j A_j - A_i\Big) \tag{4.6}$$

In our example, A_4 and A_1 are the outliers:

$$\arg\max_{A_1,A_2,A_3,A_4} l_1\Big(\frac{1}{4}(A_1+A_2+A_3+A_4) - A_i\Big) = A_4 \tag{4.7}$$

$$\arg\max_{A_1,A_2,A_3} l_1\Big(\frac{1}{3}(A_1+A_2+A_3) - A_i\Big) = A_1 \tag{4.8}$$

The remaining training signatures, A_1 and A_2, constitute the kNN model for NIC A that later will be used for transmitter identification.

4.6.1.2 Identification phase. In the simplest scenario, a PARADIS/PN-analysis sensor simply demodulates all wireless frames, extracts radiometric signatures and converts them to vectors, which are then sent to a server for identification along with the MAC address of the frame. The server may perform identification immediately, or delay identification until it accumulates multiple signatures known to be from the same transmitter. We, as PARADIS's authors, assumed that identification is always performed on groups of frames, that we call bins, that could have one or more signatures. In a realistic scenario, multiple wireless node transmits at the same time. To avoid that frames generated by different nodes are collected in the same bin, a location distinction system, as the ones we have described in Section 2.2, can be used to select and gather only frames generated by the target node. Every incoming signature in a bin is compared to all the signatures learned during training by using the l_1 function and best match is identified based on the similarity values. For example, the similarity between signatures A_1 and A_2 is 0.8, and between A_3 and A_4 is 1.2. The return value is the identity that appears most frequently among the computed best matches. In case of a tie, the identity with the greatest cumulative similarity is chosen.

4.7 Experiment

In our experiment we have chosen to test our approach, PN-analysis, that is extension and improved version of PARADIS, in comparison with PARADIS itself to understand if our phase noise analysis can improve the overall performance for wireless identification. Since we have implemented all the components (transmitter, receiver, channel, spectrum analyzer, etc..) of our experiment in a computer simulation, we have been particularly careful with the parameters choices, and we have decided to perform two separate simulations for different sets of wireless cards.

We want to underline that, when it has been possible, we have taken the values of our parameters from commercial data sheets, research papers, commercial catalogs and so on.

4.7.1 Parameters. Our original intention was to perform wireless identification on a set of wireless cards theoretically made by the same manufacturer, because it is a harder challenge. Indeed, although the standard manufacturing and quality control levels are not sufficient to eliminate all the RF impairments that uniquely characterize a NIC, network cards produced by different manufacturers are much more easily identifiable because of completely different manufacturing procedures. Moreover, we have decided to focus on few parameters to create realistic sets of networks card theoretically identical, but with small differences.

In light of this, the parameters we have chosen for characterizing different NICs are: (i) I/Q imbalance and (ii) Phase noise level.

For the I/Q imbalance values, we want to give special thanks to Dr. Vladimir Brik and his research group, the authors of PARADIS, that have shared with us some measurements regarding their experiments, allowing us to set up this parameter in a realistic way for every NIC. Since it is not our interest to understand the causes of

these impairments we do not provide a detailed description of that.

For the phase noise values, we have followed a thesis [Montoies, 2008], official data sheets and commercial catalogs for 802.11b network cards and VCO manufacturers [Crystek Corporation, , DrawCom Pty Ltd, , Skyworks,]. Since the IEEE 802.11 standard specifies maximum errors, in the modulation domain, only based on the measurement of the EVM, a phase noise constraint is not directly defined but it is implicit in the EVM that, however, gives an overall description of the general system performance, taking into account all the possible problems, RF impairments, noise and interface. Whatever, a network card with a very high level of phase noise is not able to satisfy the EVM requirement. For these reasons, even if the communication standard requirements are all satisfied, different manufactures sell products (oscillators, NICs and so on) with different average level of phase noise, at different cost. To better understand if our phase noise analysis is practicable, we have decided to perform tests on two different sets of NICs characterized by different average phase noise levels.

As described in Section 3.5.2, our phase noise model allows only to simulate flicker noise and, for this reason, we are able to simulate only a slope in the phase noise characteristic. In the data sheets, the phase noise characteristic is described by one or more values, at a max three, for different offsets; in Table 3.1 is possible to see an example.

Based on this information, the two sets of NICs are characterized by identical I/Q imbalance but with different phase noise: (i) in the first dataset, called dataset I, of NICs, the averaged phase noise is set at 71 dBc/Hz at an offset 10 kHz, (ii) in dataset II, the phase noise is 80 dBc/Hz at the same offset.

In this way, we can test our approach for two different manufactures and

different manufacturing and quality control.

Unfortunately, the documents that describe the statistical distribution of phase noise for NICs of the same manufactures are not public. Moreover, to characterize this important parameter, we have taken our decisions based on a research perform by another IIT student who designed a low-power CMOS VCO for a 2.4-GHz transmitter [Montoies, 2008]. Based on his simulation, we have found out that different network cards, theoretically identical, show different levels of phase noise with an almost normal distribution.

Based on the results of this research, we have chosen to statistically generate the phase noise for the two sets of NICs as described in Table 4.3, where the mean is equivalent to the averaged phase noise level for all the NICs in one set at an offset of 10 kHz from the carrier frequency.

Table 4.3. Phase noise characterization

	Statistical distribution	Mean [dBc/Hz] @ 10 kHz	Variance
Dataset I	Normal	-71	0.6358
Dataset II	Normal	-80	0.859

In addition, we have performed several tests changing the signal quality, acting on the SNR of the AWGN channel to understand how this parameter affects the performance.

4.7.2 Overall performance. In this Section we present the results of our tests and, subsequently, we will analyze all the details. Since we have used PARADIS as framework of our work, we can directly compare the overall performance of the two different approaches.

We have to underline that we have designed a baseband model and, moreover,

we are unable to use the frequency error metric (see Section 4.4), used by Brik. et al. in their work. Since our evaluations, in both cases, miss this metric, the performance is generally worse, for the PARADIS approach, than the ones reported on the original paper [Brik et al., 2008]. Nevertheless, this fact does not invalidate our study because our goal is to demonstrate the feasibility of our phase noise analysis for this problem or, in other words, that the phase noise metric is useable and helps to improve the identification rate. It is clear that both the approaches can only be improved, in a realistic implementation, adding the missed metric.

As we have explained in the previous section, we have created two different sets of NICs, and in Table 4.4 is possible to confront directly the overall performance for both approaches. The last line of the table presents common parameters used in both cases. They will be explained in detail later in this chapter.

Table 4.4. Comparison of PN-analysis and PARADIS

	NICs	Averaged PN Level @ 10 kHz [dBc/Hz]	Average Error Rate for PARADIS	Average Error Rate for PN-analysis
Dataset I	40	-71	8%	3.625%
Dataset II	40	-80	3.25%	0.88%
training size = bin size = 20 frames, SNR for training/identification phase = 20 dB				

In both cases, the AER is improved by the PN-analysis in comparison with PARADIS. Since PN-analysis is an extension of the other approach we can infer that the improvement is addressable to the phase noise measurement, that is an useable metric for this problem and that this approach is feasible.

4.8 Results for Dataset I

Before any other evaluation, we want to understand more in depth how the phase noise measurement is affected by the signal quality and what changes in our PN-analysis.

To be consistent with the tests we will show after in this section, the phase noise is calculated on a transmitted signal composed by 20 IEEE 802.11 frames, at $f_m = 10$ kHz offset from the carrier.

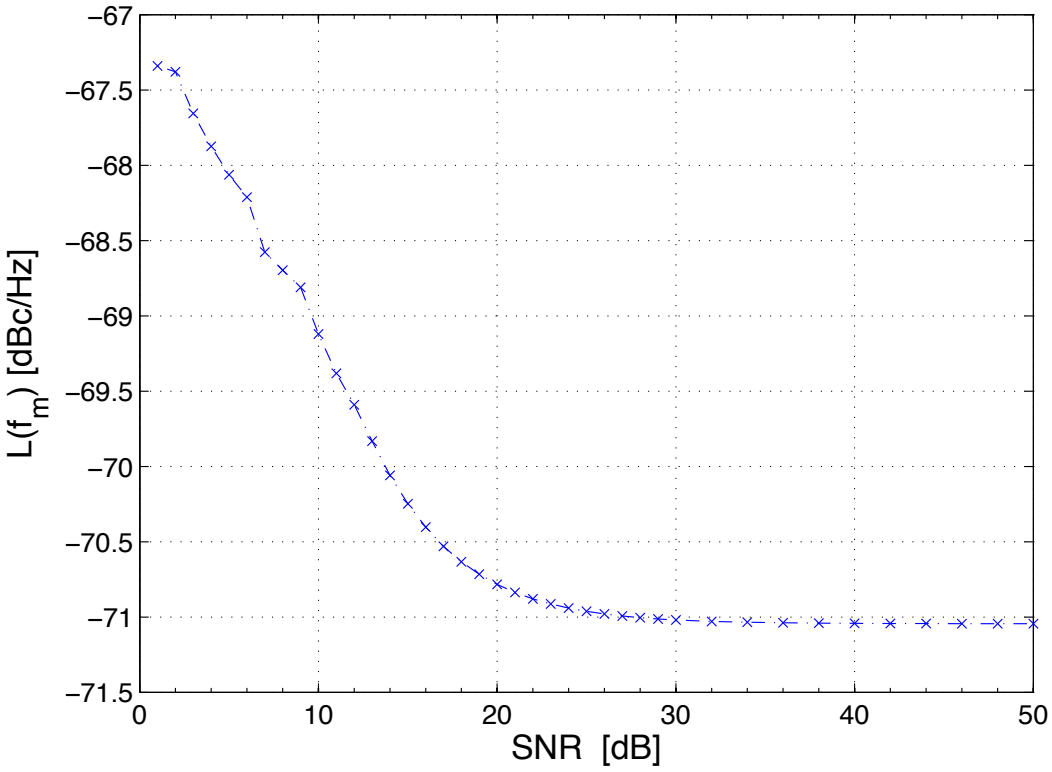


Figure 4.6. Phase noise measurements with respect to SNR for dataset I

In Fig. 4.6 it is possible to see that as the SNR decreases, the estimation of phase noise is less accurate. This is because the phase noise measurement is based on the detected symbols that are assumed, because it is the only information we have, to be equal to the transmitted ones. Unfortunately, low SNR creates troubles to this method because the lower the SNR, the more probable a wrong decision by the receiver is.

Very high SNR values are not realistic in reality. We have decided to set the maximum SNR achievable at 20 dB, which also guarantees a good phase noise

measurement; especially for the training phase, i.e. when the network administrator has the availability of the NICs to generate the signatures, an SNR around 20 dB perfectly matches realistic conditions because, in this scenario, the targeted NIC is nearby the PARADIS/PN-analysis sensor and the wireless link is almost not affected at all by interference or noise.

For the identification phase, due to the characteristics of the environment, for instance the number of wireless nodes transmitting at the same time or interference from other RF devices that work at similar frequency, a high level of SNR is not assured. For this reason we have performed several tests with different levels of SNR for this phase. It is possible to see how the performances of the two approaches change with different signal quality levels in Fig. 4.7.

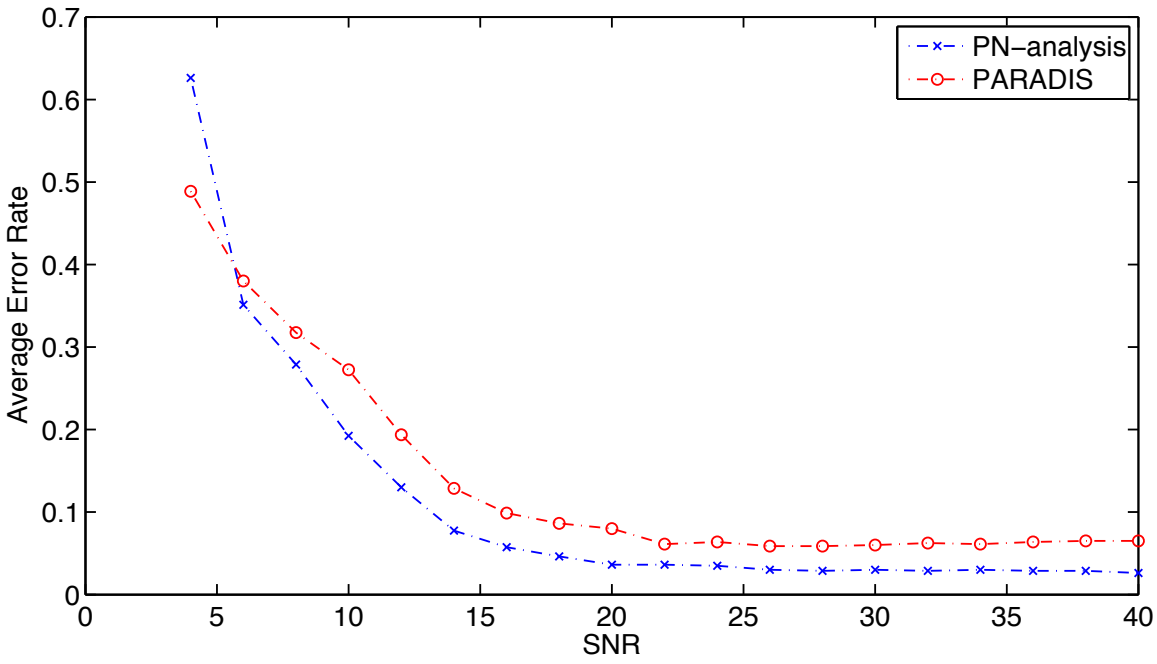


Figure 4.7. Average Error Rate with respect to SNR for dataset I

4.8.1 Parameter tuning. We now discuss the empirically established parameters of the optimal model. Note that performance numbers reported in the following sections are not meant to quantify and compare optimal performance. The tuning

models were suboptimal in at least one of their parameters.

4.8.1.1 Training set size. We evaluated the effect of training set size on effec-

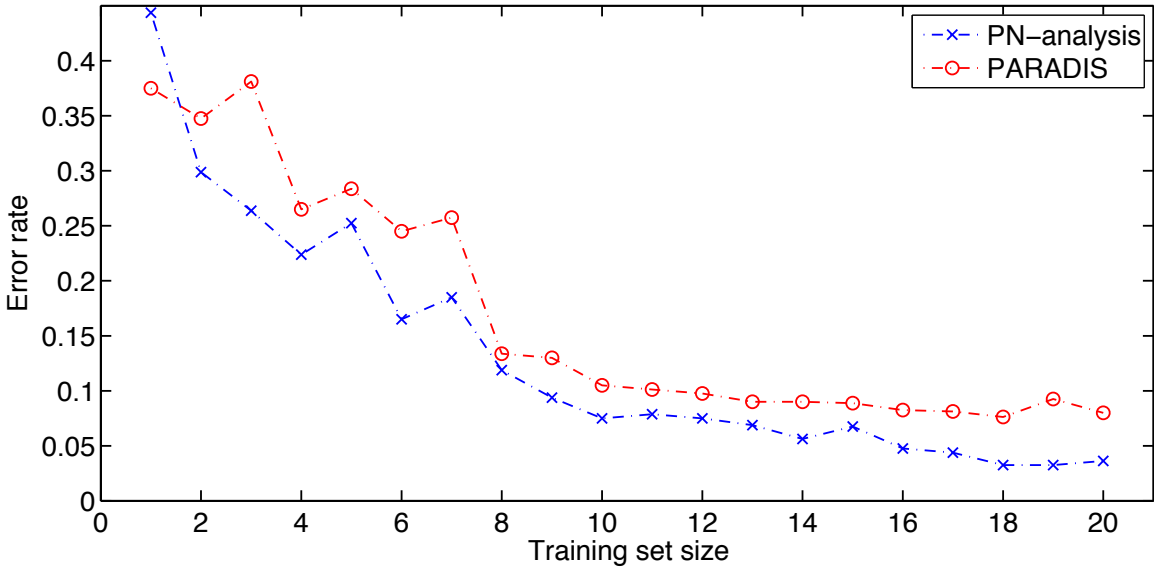


Figure 4.8. Effect of training set size on accuracy for dataset I

tiveness of the two approaches by performing classification on the dataset varying the training set size but keeping the bin size at 20 frames, which is the optimal value as we will see shortly. We will make two observations on the effect of training set size on accuracy, as shown in Fig. 4.8. The first thing to notice is that the error rate for PN-analysis is always lower than the PARADIS's one. The only exception is with a training set size of 1 frame, but the performance of both cases are not sufficient and, moreover, we need more frames to perform an accurate identification. The performance is sufficiently good with 10 frames but, since during the training phase we are operating in particularly good conditions because we have physically the availability of the NICs, we decide to keep a training set size of 20 frames, that are sufficient to perform identification with a error rate of 8% for PARADIS and 3.625% for PN-analysis.

4.8.1.2 Tuning bin size. Relatively few applications require per-frame identi-

fication. At the same time, it is often possible to infer which frames originated at the same NIC. For example, 802.11 acknowledgements and retransmissions can conveniently boost the number of frames that are highly likely to have come from the same transmitter, if it is not sufficient we can use a location distinction system.

In this section we present the relationship between bin size and accuracy keeping the training set size on 20 frames, i.e. the optimal size as we have just seen. Fig. 4.9 illustrates the effect of binning on average misclassification rate. For a bin size lower than 10, PARADIS performs better. On the other hand, the results are tipped over from greater bin size. Indeed, the performance of PN-analysis gets better with more data. The phase noise is a random phase fluctuation, and the estimation is more accurate when we perform it on more frames, mitigating detection errors.

Even in this case, we decide that the optimal bin size is 20 frames, that is a good trade-off between the time to capture them and the result's accuracy. In this case, PARADIS experienced an AER of 8% while PN-analysis has 3.625%.

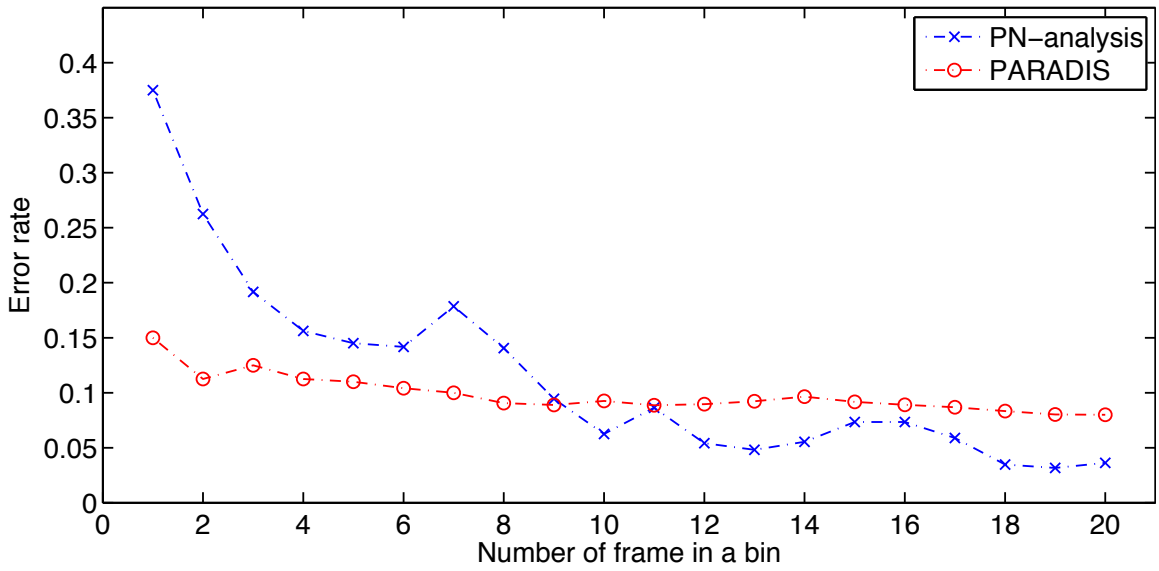


Figure 4.9. Effect of bin size on accuracy for dataset I

4.8.1.3 Individual NICs performance. To perform this analysis we have set

SNR for training and identification phase at 20 dB, and we kept training set size and bin size at 20 frames. In Figures 4.10-4.11, it is possible to confront the FRR for the two different approaches.

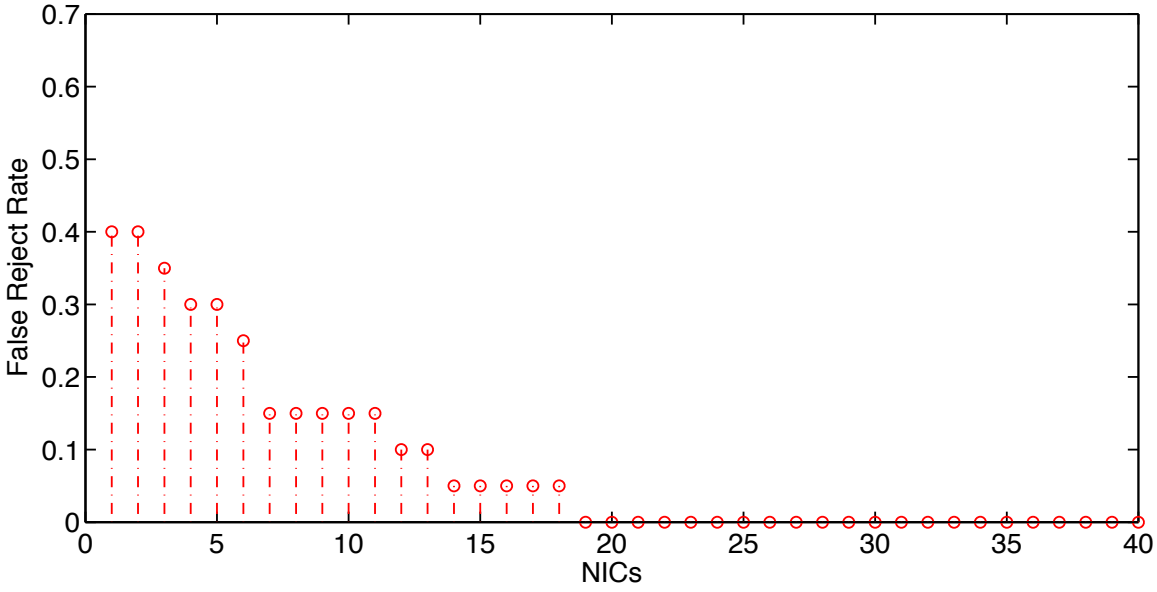


Figure 4.10. False Reject Rate for PARADIS for dataset I

The average error rate for PARADIS was 8% in these conditions. However, Figure 4.10 shows that 45% (18 cards) of all the cards experimented non-zero FRR, in particular equal or greater 5%. Among them, almost 32.5% (13 cards) have FRR greater or equal to 15%. Still, 27 NICs (67.5%) only saw 5% or lower FRR.

PN-analysis, shown in Figure 4.11, had an AER of 3.625% and experimented only 7 NICs (17.5%) with non-zero FRR. Only 2 of them (5% of the totality) tower that of the majority with FRR greater than 15% with values of 35% and 65%, respectively. In this case, 90% of the population (36 NICs) had FRR lower than 10%.

Similarly, Figures 4.12-4.13 show the performance with respect to the WCS.

As in the previous case, 18 cards (45%) experimented non-zero WCS for PARADIS and 32.5% (13 NICs) had WCS greater or equal 10%. Only 7 NICs (17.5%)

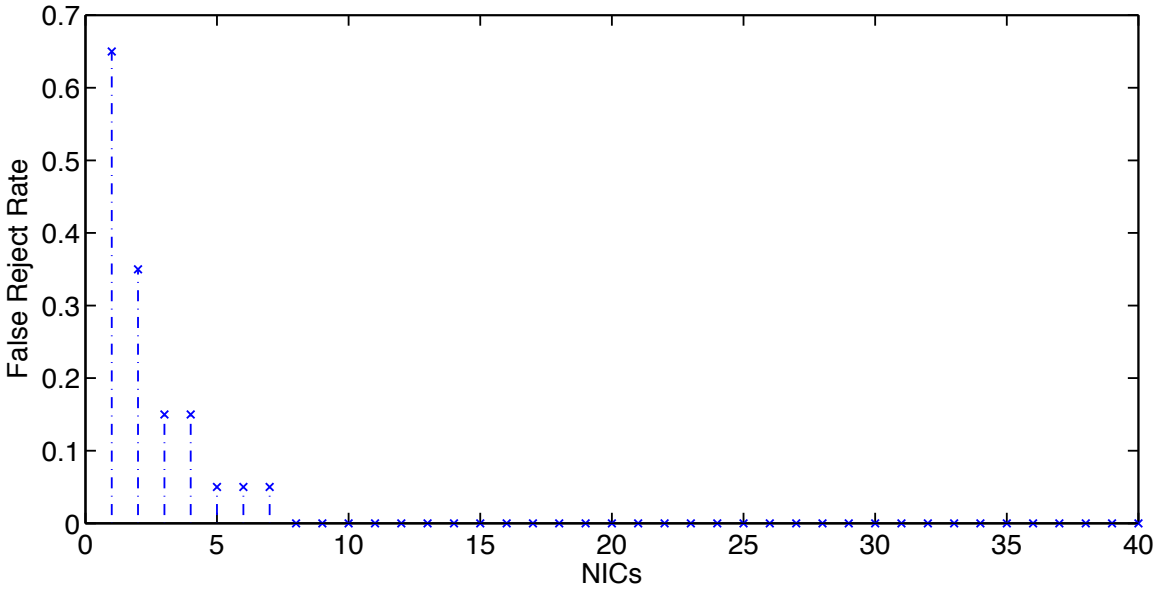


Figure 4.11. False Reject Rate for PN-analysis for dataset I

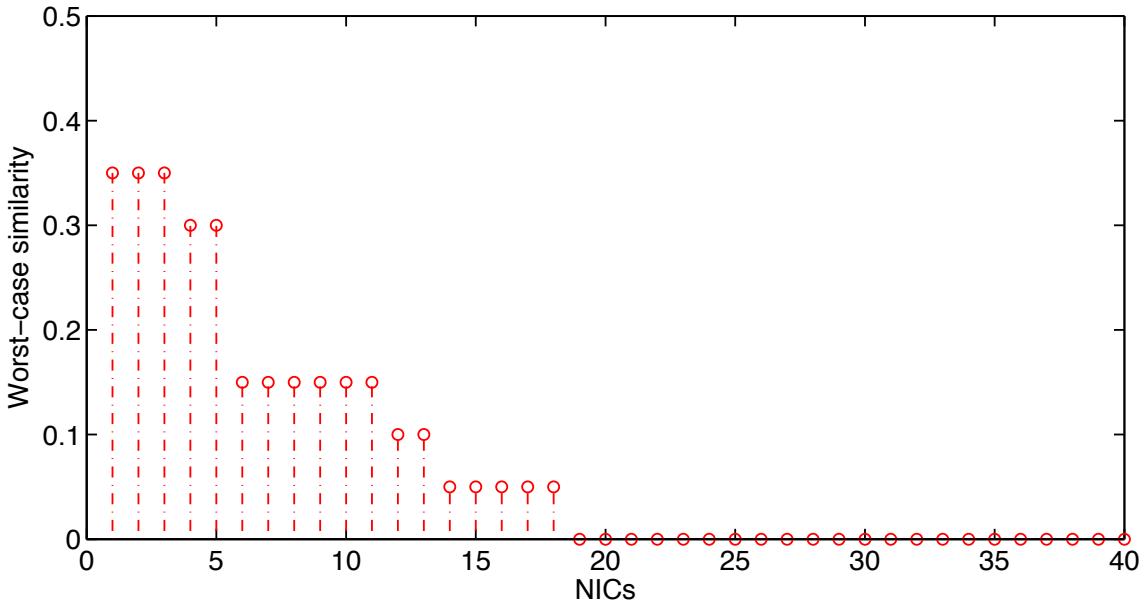


Figure 4.12. Worst-Case Similarity for PARADIS for dataset I

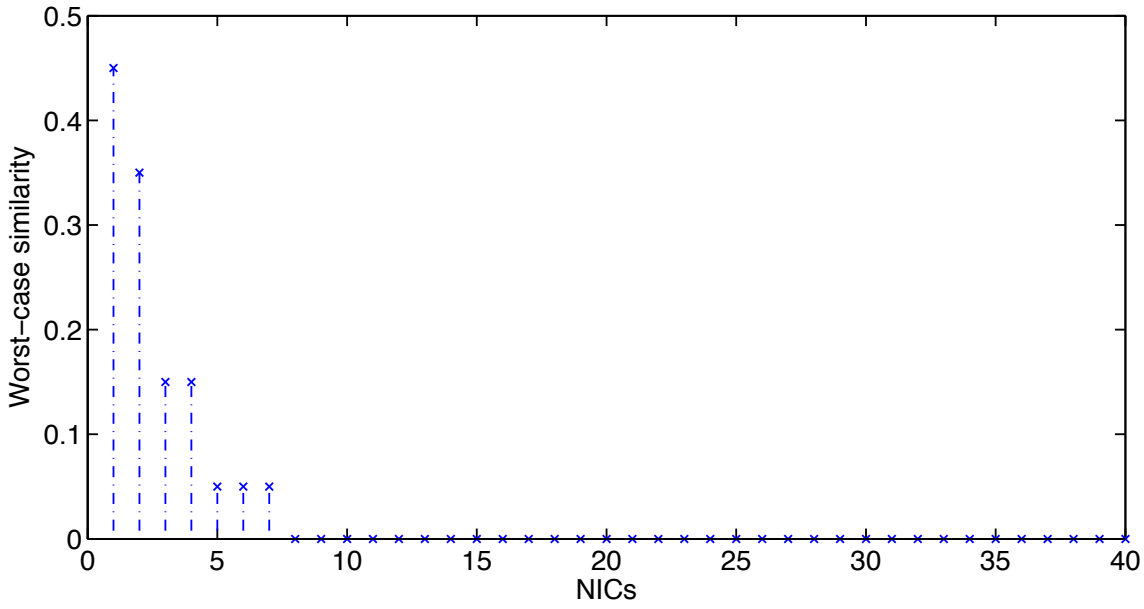


Figure 4.13. Worst-Case Similarity for PN-analysis for dataset I

had non-zero WCS for PN-analysis, and only 10% (4 NICs) had it greater or equal than 10%. Interestingly, these are not the same NICs that caused trouble with PARADIS and PN-analysis for both metrics, but we notice a correlation between FRR and WCS in both cases. Indeed, when a NIC experiments a high false reject rate, i.e. the system likely links frame signatures to a wrong NIC among the population, also has an high worst-case similarity because the misclassification is not random but it is connected to the fact that few NICs have similar signatures.

PN-analysis consistently outperforms PARADIS, whose performance, nevertheless, is sufficient to be useful in certain applications, especially involving constrained computation resource, or implementation in hardware.

We can make several observations based on this data, and we focus particularly in the performance of PN-analysis. First, the majority of NICs were not able to masquerade as another. With the exception of two NIC that did very badly in both the metrics (between 30-60% for WCS and FRR), the maximum WCS for PN-analysis

is 15% (for 2 NICs) or lower. Indeed, only about 1 out of 10 frames sent by the perpetrator was mistaken as coming from the victim, making it very unlikely to avoid detection. At the same time, similarity was under 10% for 90% of the population, most of which was not affected at all. Another observation is that very few NICs of the 40 transmitters in our dataset accounted the majority of all misclassifications.

We want to underline how PARADIS's results came out from our simulation are very close to the ones published by Brik et al. in their work [Brik et al., 2008]. This fact confirms the validity of our model that behaves similarly to real experiments done in laboratory by another research group.

4.9 Results for Dataset II

We now repeat, for dataset II, the analysis just done for dataset I. All the assumptions and the parameter choices we have done before are still valid because we want to test and compare our method only with a difference dataset, that differs from the other by the level of phase noise in the network cards, now set at 80 dBc/hz as we have shown in Table 4.3.

As we have notice for dataset I, the phase noise measurement at $f_m = 10$ kHz, shown in Fig. 4.14 improves with the SNR. In this case, we underline that the phase noise estimation is not as accurate as before until SNR equals 30 dB. For example, for SNR = 20 dB, the phase noise estimation is -78.4 dBc/Hz. Unfortunately, as we have stated before, too high SNRs do not respect realistic environmental conditions. For the same motivations as before we set the maximum SNR achievable at 20 dB.

In our opinion, this kind of error will not be present in an experiment with real NICs and spectrum analyzers because the latter is designed for these kind of measurements. Even if our simulation model is theoretical, we only apply the definite formula of phase noise, without any optimizations or further considerations. As it is possi-

ble to see here [Agilent Technologies, 2005], [Agilent Technologies, 2005], spectrum analyzers are very complicated tools, specifically designed to consider and mitigate the effects of several working conditions and errors for different offsets. Furthermore, we think that in a real experiment the phase noise measurements and analysis can perform with a good accuracy in the majority of the working conditions, based on the good performance on spectrum analyzers for this kind of measure.

The phase noise calculations represented in Fig. 4.14 have been taken over 20 frames.

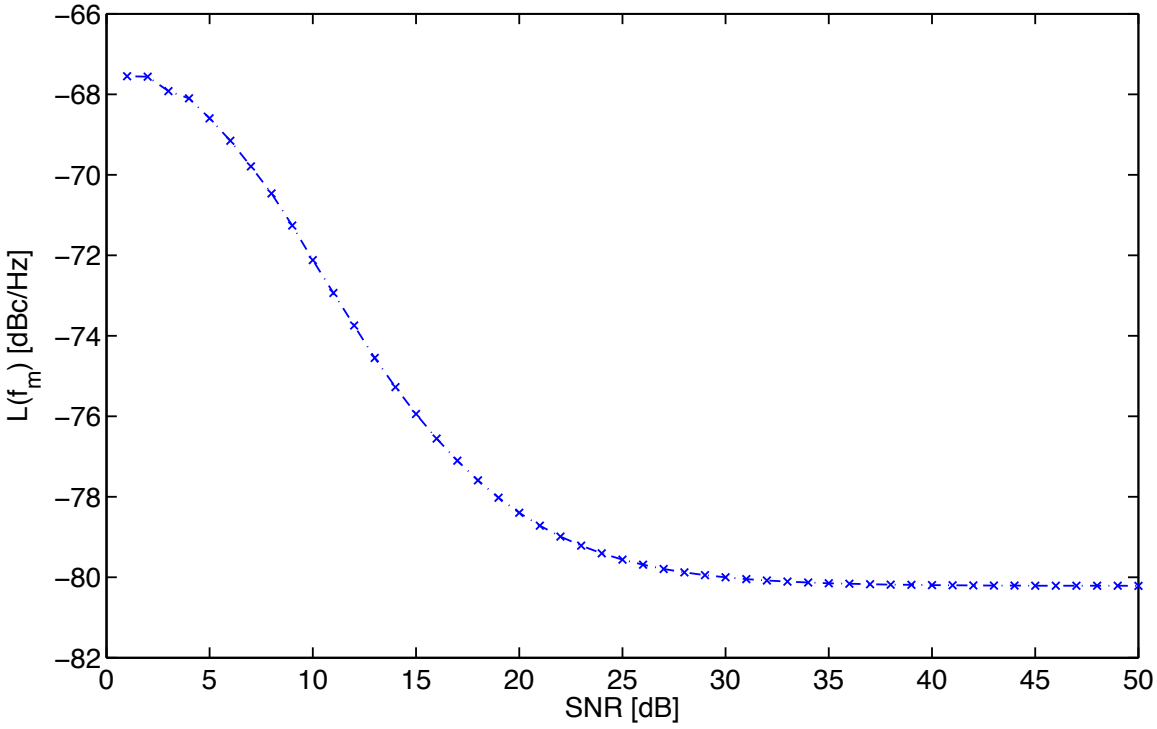


Figure 4.14. Phase noise measurements with respect to SNR for dataset II

Fig. 4.15 shows the performance of the two approaches with different SNRs. As before, the training set size and the bin size are set to 20 frames, and the SNRs during training and identification phase are both set to 20 dB. In this case, it is interesting to notice how PN-analysis does not outperform PARADIS until the SNR is high enough, with a starting point around 16.5 dB. We inspected thoroughly the

experimental data and we noticed that, for lower phase noise levels, the measurements are less accurate, especially for low SNRs. Because of that, the system is less capable to quantify the differences of PN between NICs and, moreover, the classifier works worse.

When the SNR is around 20 dB the performance reach its maximum (i.e. the minimum in term of AER).

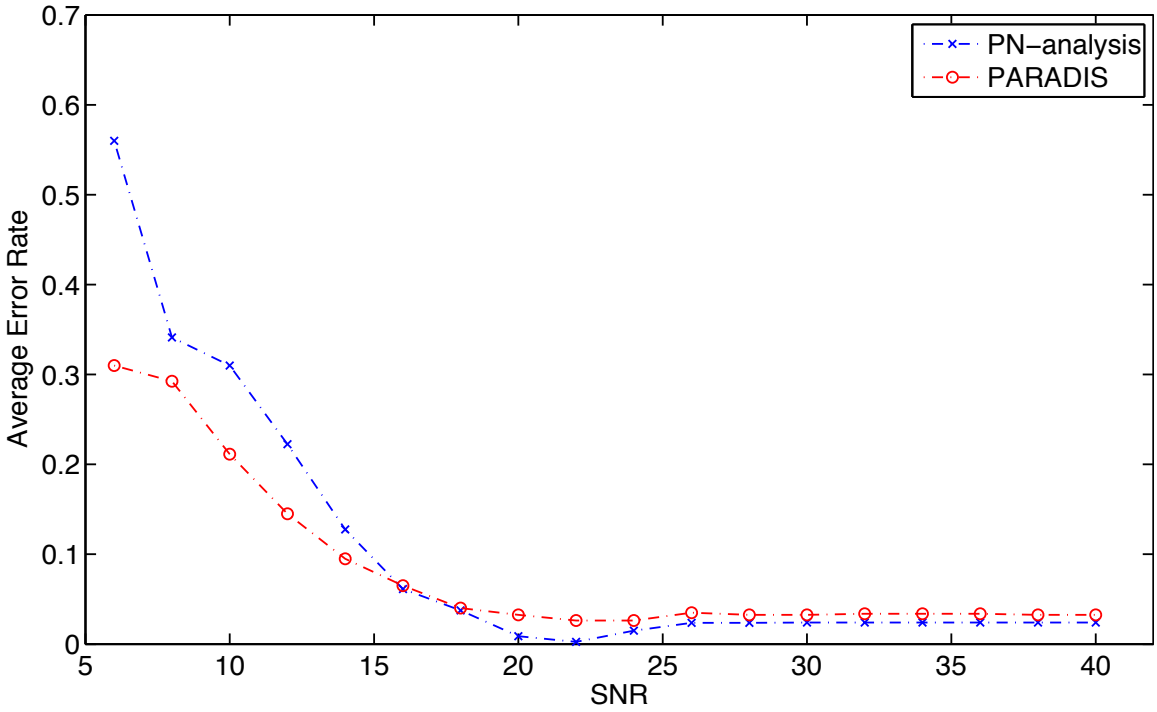


Figure 4.15. Average Error Rate with respect to SNR for dataset II

4.9.1 Parameter tuning.

4.9.1.1 Training set size. Fig. 4.16 shows how the two approaches perform with different training set sizes. In this case, PN-analysis outperforms PARADIS only when we use more than 10 frames for the training phase, that means the system selects and conserves 5 signatures for every NIC, while the others are discarded because outliers. After 8 frames, the performance for PARADIS reaches their best result, as it is reported also in the original paper [Brik et al., 2008] in the real experiment. For

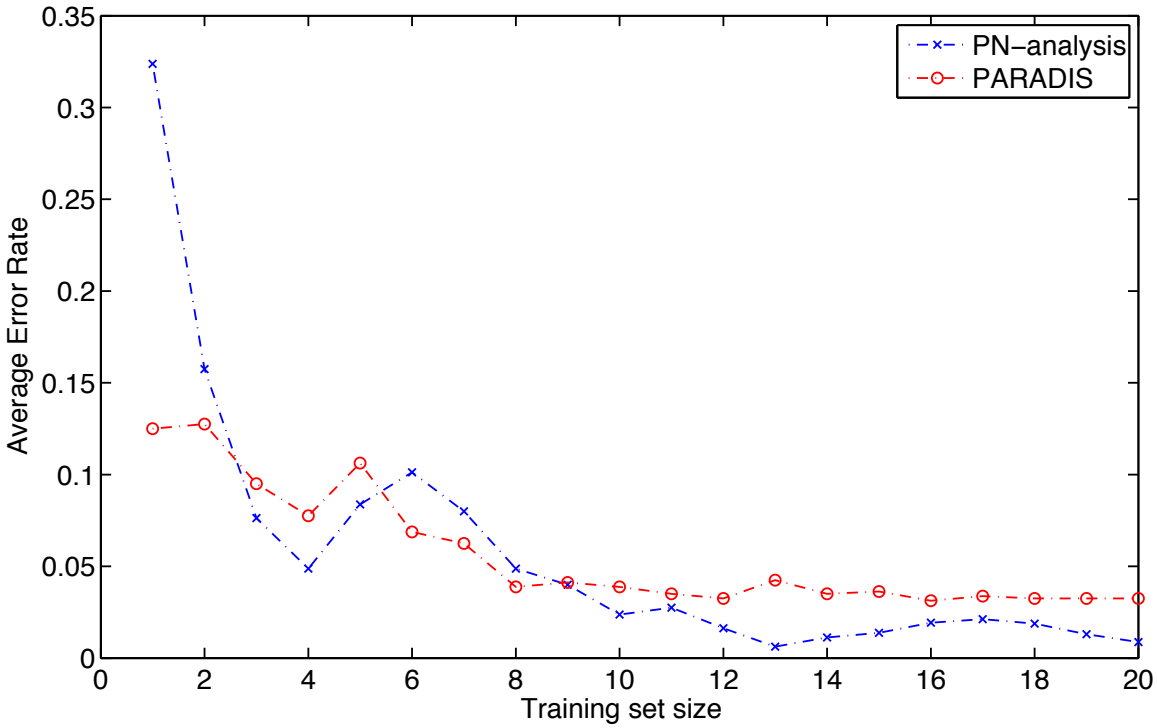


Figure 4.16. Effect of training set size on accuracy for dataset II

PN-analysis, the performance levels off after 10 frames. Therefore, just a few frames capture the radiometric identity of a NIC.

We notice that, with a sufficient number of frames that guarantees a good AER, our approach works better than PARADIS. In fact, between 10 and 20 frames PARADIS experienced an AER of 3% on average, while PN-analysis was around 1-1.2%.

4.9.1.2 Tuning bin size. Fig. 4.17 illustrates the effect of binning on average misclassification rate. In this case, PN-analysis does not improve PARADIS's results until the bin size is lower than 12. As it is reported in the original paper by Brik et al. PARADIS does not take advantage of a bin size greater than 4. On the other hand and due to the phase noise measurements that require more data to work well, PN-analysis needs a bin size greater or equal to 12 to outperform PARADIS. The

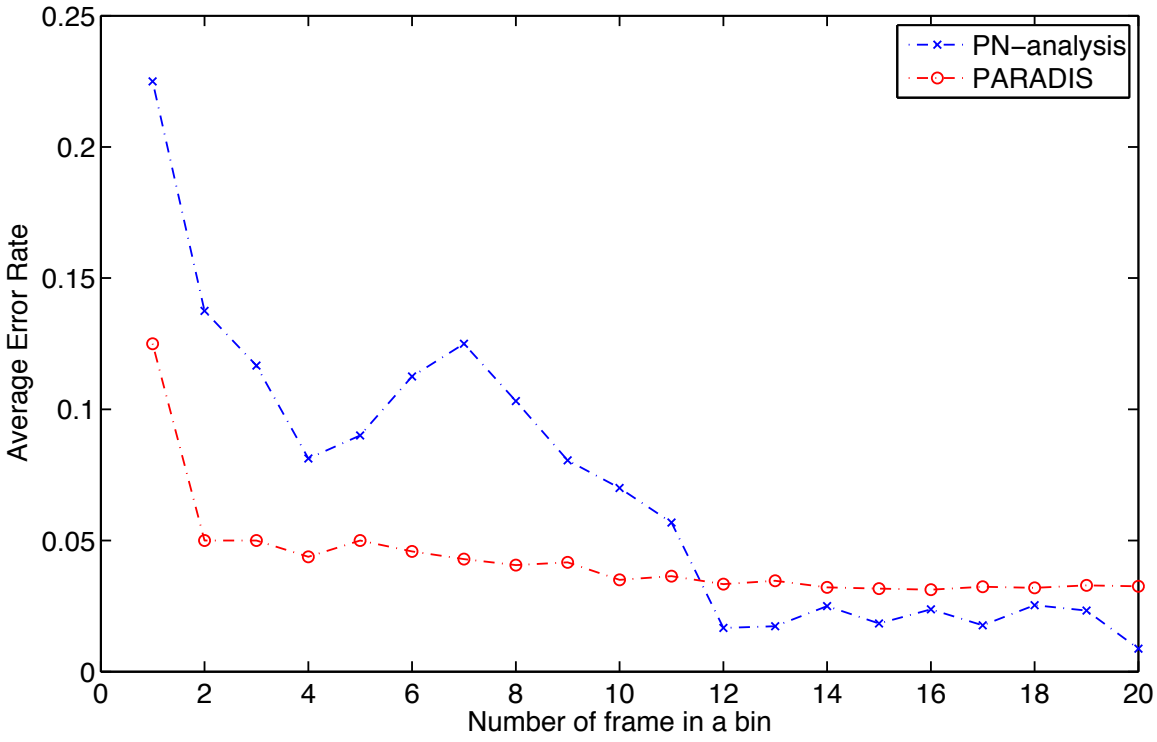


Figure 4.17. Effect of bin size on accuracy for dataset II

best performance are, anyway, reached with a bin size of 20 frames.

4.9.1.3 Individual NICs performance. To perform this analysis, we have set SNR for training and identification phase at 20 dB, and we kept training set size and bin size at 20 frames. In Figures 4.18-4.19, it is possible confront the FRR for the two different approaches.

The average error rate for PARADIS was 3.25% in these conditions. Figure 4.18 shows that 20% (8 cards) of all the cards experimented non-zero FRR, equal or greater 5%. Among them, 12.5% (5 cards) have FRR greater or equal 10%. Still, 32 NICs (87.5%) only saw 5% or lower FRR.

PN-analysis, in Figure 4.19, experimented only 3 NICs (7.5%) with non-zero FRR and only 2 of them (5% of the totality) with an FRR of 15%. In this case, 95% of the population (38 NICs) had FRR lower than 10%.

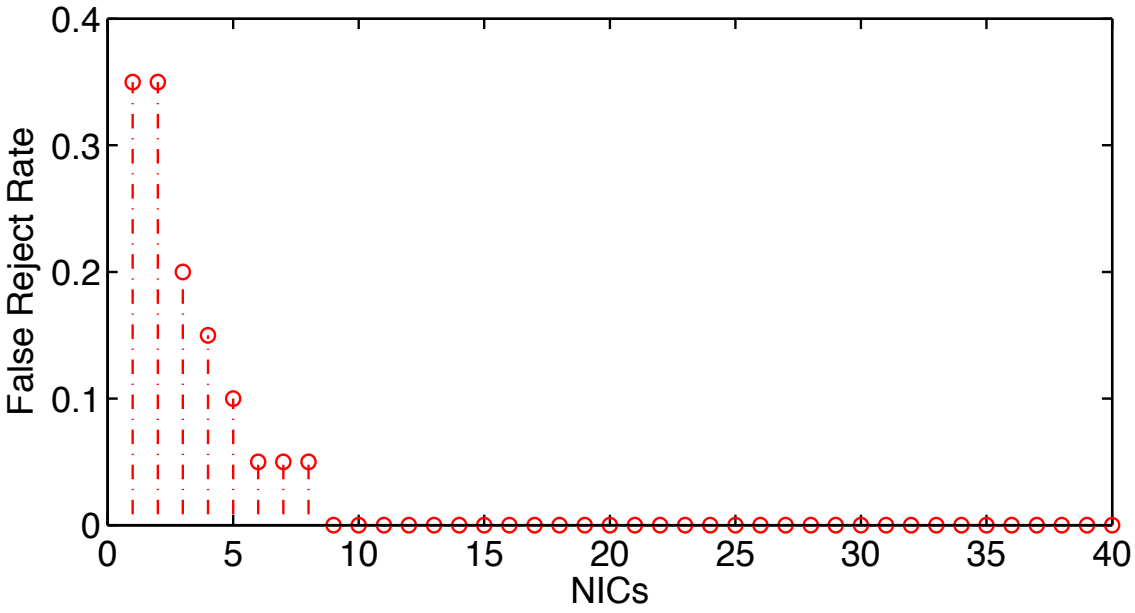


Figure 4.18. False Reject Rate for PARADIS for dataset II

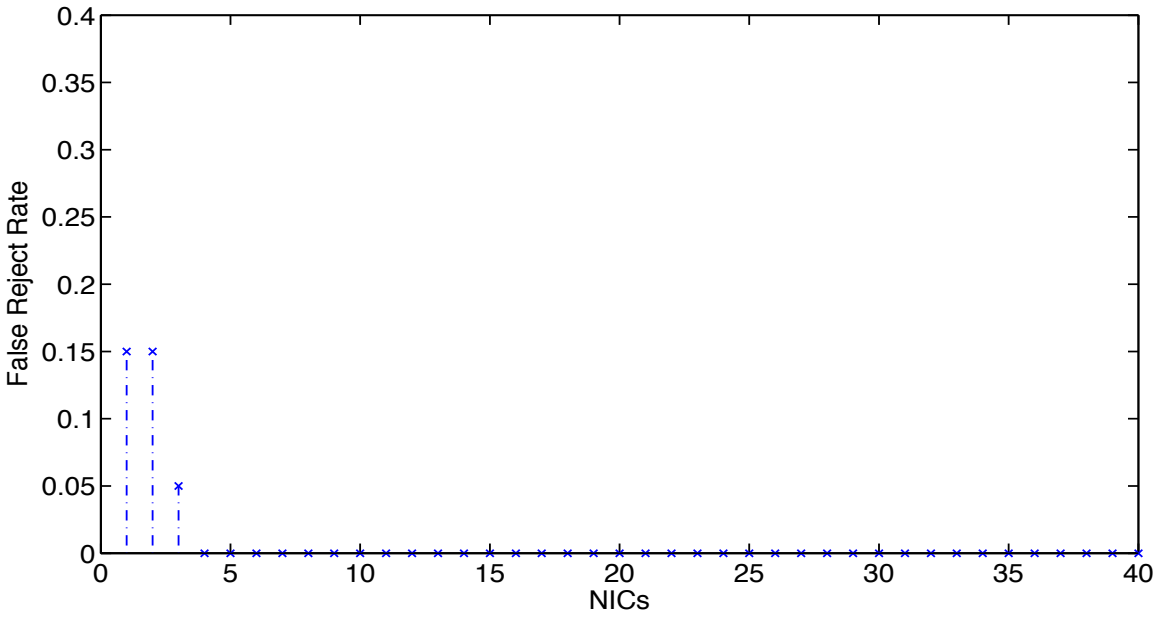


Figure 4.19. False Reject Rate for PN-analysis for dataset II

Similarly, Figures 4.20-4.21 show the performance with respect to the WCS. As in the previous case, 8 cards (20%) experimented non-zero WCS for PARADIS and 12.5% (5 NICs) had WCS greater or equal 10%. Only 3 NICs (7.5%) had non-zero WCS for PN-analysis, and only 10% (4 NICs) had it greater or equal than 10%. Again, these are not the same NICs that caused trouble with PARADIS and PN-analysis for both metrics but we noticed a strong correlation between FRR and WCS in both cases, separately. We have already discussed about this with the previous dataset also.

PN-analysis consistently outperforms PARADIS, whose performance, nevertheless, is sufficient. Since the results are similar to the other dataset, we suggest to review the consideration in the previous section.

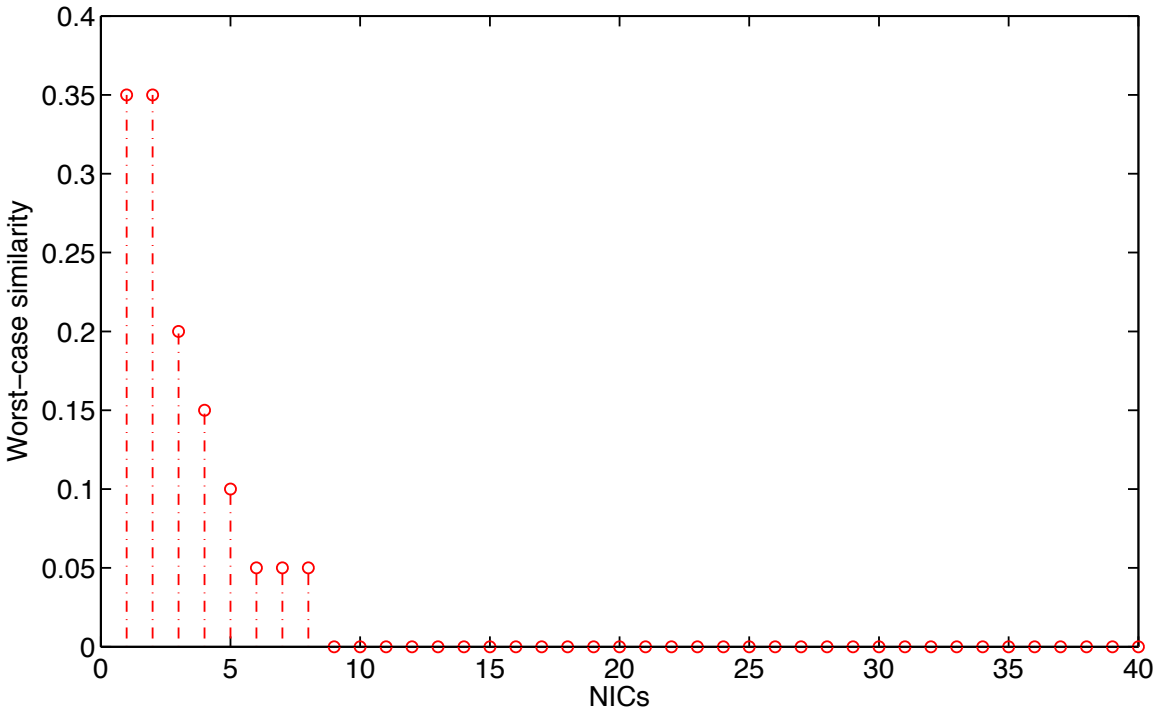


Figure 4.20. Worst-Case Similarity for PARADIS for dataset II

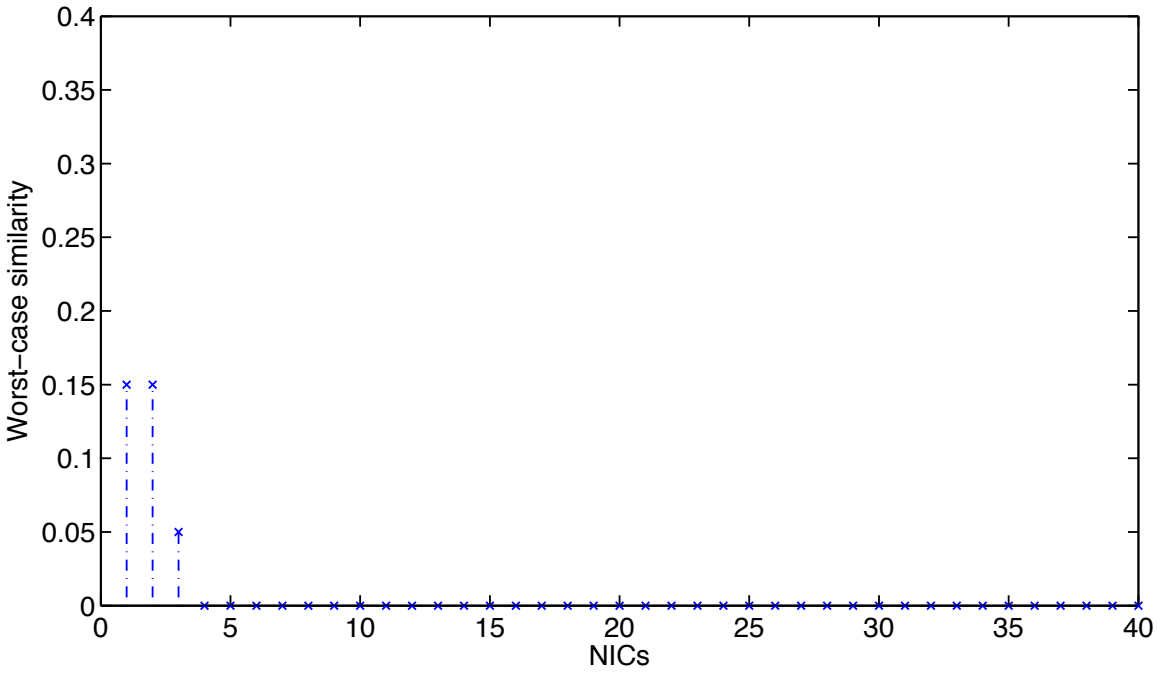


Figure 4.21. Worst-Case Similarity for PN-analysis for dataset II

CHAPTER 5
CONCLUSION

5.1 Summary

In this thesis we have addressed the issue of wireless NIC identification at the physical layer. The main idea of wireless identification techniques is to identify wireless transmitter devices based on minor artifacts in their emissions that are produced by idiosyncratic hardware properties of individual NICs.

We used as framework of our research a brand new approach for WIS, called PARADIS, that already was able to identify NICs and detect wireless impostors with very good accuracy. Unlike the previous state-of-art techniques, PARADIS defines a signal's signature in terms of structure imposed by the modulation scheme; in substance, PARADIS based its analysis averaging modulation errors. We chose this approach because it greatly simplifies the problem, allowing us to extend it and test our idea very quickly.

Furthermore, our main goal was to demonstrate, using computer simulations, the feasibility to exploit the phase noise, present naturally in oscillators, to discriminate between different NICs. The performance of PARADIS in our computer simulations, closely follows the real one published by Brik et al. in their paper, indirectly validating our model.

This thesis will be part of a future research of an IIT PhD student, who will test our idea in laboratory. In fact, before allocating the budget to buy all the components (wireless cards, spectrum analyzers, etc..), which are very expensive, we wanted to prove the practicability of the phase noise analysis.

Comparison of PARADIS and our approach, that we termed PN-analysis, involved two different dataset of 40 NICs for different phase noise values. We have per-

formed several tests to understand how our analysis is affected by different SNRs and how the performance changes. Generally, our technique outperformed the PARADIS approach, and proved to be useable and feasible to carry out wireless identification. More specifically, PN-analysis has reported an AER of 3.625% and 0.88%, respectively for dataset I and dataset II, in comparison with PARADIS that had figures of 8% and 3.25%, respectively for the two datasets. PN-analysis also shows a significantly increase in the number of NICs not experiencing any misclassification, and an important reduction for the two metrics FRR and WCS for every NIC. We have to notice that PN-analysis generally requires more frames than PARADIS because the estimation of phase noise is more accurate when we realize it with more data.

A big disadvantage of our approach is that it requires specific hardware, such as a spectrum analyzer or a special system to realize phase noise measurements, and so a PN-analysis sensor will be more complex than a PARADIS one, which is basically a normal receiver with advanced signal processing features but without any additional and expensive hardware. For this reason, we think that our approach can be worthy enough to be implemented in those situations where security is a paramount requirement.

Since spectrum analyzers are specific tools for measurements taken in the frequency domain, we suffice that their accuracy could be even higher than in our simulation, because they deeply elaborate the signal to mitigate errors and interferences.

5.2 Applications

We now consider some important applications that may be realized by our technique. Note that we will not be elaborating extensively on the applications as we consider that our most important contribution is to proved the usability of phase

noise for wireless identification systems. We believe that our technique can be tuned to suit any application where identity resolution in WLAN is a problem.

At this time, no identification mechanism that relies solely on physical layer measurements of intercepted analog signals achieve perfect accuracy because of environmental noise, and our approach is not an exception. Consequently, we suggest physical layer identification systems, such as PARADIS or our PN-analysis, as a secondary security perimeter that detects breaches in the primary perimeter established, for instance, using cryptographic mechanisms. Although the following applications are technically feasible, we recognize the critical need to preserve the privacy of users.

5.2.1 Network Forensics. As reported by [Desmond et al., 2008], there could be problems when network logs are used in a legal proceedings. At present, network logs are recorded such that the notion of virtual identity is tied to IP addresses, MAC addresses and user accounts. These data can be easily altered or directly spoofed by an attacker without leaving any trace. Thus intrusion detection records and network traces may in fact point to the wrong culprit, instead of identifying the right attacker. Again, this problem can be resolved if wireless device fingerprints can be used as an independent mechanism to identify unique devices.

5.2.2 Spoof Detection. Several groups of students have already demonstrated how inadequate are the security techniques for wired networks applied to a wireless one [Borisov et al., 2001], [Arbaugh et al., 2002], [Walker et al., 2000].

At present WEP can be cracked and even with WPA, management and control frames remain susceptible to MAC spoofing/replay/masquerading attacks due to lack of authentication and encryption for such frames. It is straightforward that a possible application of our wireless identification system is to protect wireless networks from spoofing or impersonation attacks. This requires, for example, the maintenance of a

white-list of allowed devices, that associates the device's MAC address to its signature. An alien device that attempts to masquerade using a valid MAC address will be picked up through device fingerprinting. In stark contrast with identification based on MAC address, our technique would prove to be much more challenging to evade.

5.2.3 Detection of compromised keys. Detection of compromised keys is one of the hardest challenges in wireless networks. A straightforward solution could be the use of a physical-layer identification system, like ours. If an unauthorized entity gains network access using a stolen a secret key, a network administrator has no chance to identify it. On the other hand, using a physical-layer identification system, even if an (internal) attacker holds an authentic cryptographic keys, he will not be able to authenticate itself to the network with his own device unless he replicates or steals the wireless device containing that particular signature recognized by the WIS.

5.2.4 Detection of wormhole attacks. Wormhole attacks are usually associated to multi-hops or ad-hoc networks In a wormhole attack [Hu et al.,], an attacker forwards packets received at one point of the network to another point that is usually multiple hops away. Two attackers, positioned at the respective point, create a tunnel in which packets are transmitted faster because they avoid a normal multi-hops route (e.g., through use of a single long-range directional wireless link or through a direct wired link to the colluding node).

Due to the nature of wireless transmission, the attacker can create a wormhole even for packets not addressed to itself, since it can overhear them on the wireless channel and tunnel them to the colluding node at the other end of the wormhole.

The wormhole attack can form a serious threat especially against many ad hoc network routing protocols and location-based wireless security systems. For example, most existing ad hoc network routing protocols, without some mechanism to defend

against the wormhole attack, would be unable to find routes longer than one or two hops, severely disrupting communication.

A wormhole attack is among the most difficult attacks to detect because it can be executed exclusively by external attackers and because the information in the packets does not need to be changed for the attack to work. That means that even encrypted and signed messages can be subject to a wormhole attack.

Physical-layer identification systems, such as [Rasmussen, 2007], PARADIS or PN-analysis helps identifying the attacker's device (intruder) when trying to forward packets, because they can detect differences in the physical characteristics of the transmitted signal.

5.2.5 Sybil attack. As reported in [Parno et al., 2005], physical-layer identification can be used to prevent Sybil [Newsome et al., 2004] and node replication (cloning) [Parno et al., 2005] attacks. In the Sybil attack, the attacker gives several identities to the same sensor node with the purpose to fool the routing and data aggregation in the network. The replication attack consists of assigning the same (legitimate) identity to several nodes. With a WIS in place, and given the difficulty of compromising the identification, these attacks can be successfully prevented.

BIBLIOGRAPHY

[Agilent Technologies, a] Agilent Technologies. Agilent 802.11 a/g Manufacturing Test Application Note: A Guide to Getting Started. *Application note*, (pp. 1308–3).

[Agilent Technologies, b] Agilent Technologies. Making 802.11g transmitter measurements. *Application note*, (pp. 1380–4).

[Agilent Technologies, c] Agilent Technologies. RF Testing of WLAN products. *Application note*, (pp. 1380–1).

[Agilent Technologies, 2001] Agilent Technologies (2001). Wireless Test Solutions.

[Agilent Technologies, 2005] Agilent Technologies (2005). Spectrum analysis basics. *Application Note*, 150.

[Arbaugh et al., 2002] Arbaugh, W., Shankar, N., Wan, Y., & Zhang, K. (2002). Your 802.11 wireless network has no clothes. *IEEE Wireless Communications*, 9(6), 44–51.

[Barbeau et al., 2006] Barbeau, M., Hall, J., & Kranakis, E. (2006). Detecting impersonation attacks in future wireless and mobile networks. *Lecture Notes in Computer Science*, 4074, 80.

[Barton & Leonov, 1997] Barton, D. & Leonov, S. (1997). *Radar technology encyclopedia*. Artech House.

[Borisov et al., 2001] Borisov, N., Goldberg, I., & Wagner, D. (2001). Intercepting mobile communications: The insecurity of 802.11. In *Proceedings of the 7th annual international conference on Mobile computing and networking* (pp. 180–189).: ACM New York, NY, USA.

[Bria et al., 2001] Bria, A., Gessler, F., Queseth, O., Stridh, R., Unbehaun, M., Wu, J., Zander, J., & Flament, M. (2001). 4th-generation wireless infrastructures: scenarios and researchchallenges. *IEEE [see also IEEE Wireless Communications] Personal Communications*, 8(6), 25–31.

[Brik et al., 2008] Brik, V., Banerjee, S., Gruteser, M., & Oh, S. (2008). Wireless device identification with radiometric signatures. In *Proceedings of the 14th ACM international conference on Mobile computing and networking* (pp. 116–127).: ACM New York, NY, USA.

[Clarke & Hess, 1971] Clarke, K. & Hess, D. (1971). *Communication circuits: analysis and design*. Addison-Wesley, Reading, Mass.; London.

[Crystek Corporation,] Crystek Corporation. *RF Products Guide - Fall 2008*, www.crystek.com/microwave/spec-sheets/rfproductguide.pdf edition.

[Dasgupta et al., 2003] Dasgupta, D., Gomez, J., Gonzalez, F., Kaniganti, M., Yallapu, K., & Yarramsetti, R. (2003). MMDS: Multilevel Monitoring and Detection System. In *Proceedings of the 15 th Annual Computer Security Incident Handling Conference, Ottawa, Canada*.

[Desmond et al., 2008] Desmond, L., Yuan, C., Pheng, T., & Lee, R. (2008). Identifying unique devices through wireless fingerprinting. In *Proceedings of the first ACM conference on Wireless network security* (pp. 46–55).: ACM New York, NY, USA.

[DrawCom Pty Ltd,] DrawCom Pty Ltd. *2008 Cattallogue*, www.drawcom.com.au/.../drawcom_issue9_coaxial_62-65.pdf edition.

[Dudani, 1991] Dudani, S. (1991). The distance weighted k-nearest neighbor rule. *Nearest neighbor (NN) norms: nn pattern classification techniques*.

[Ellis & Serinken, 2001] Ellis, K. & Serinken, N. (2001). Characteristics of radio transmitter fingerprints. In *Radio Science*, volume 36 (pp. 585–597).

[Faria & Cheriton, 2006] Faria, D. & Cheriton, D. (2006). Detecting identity-based attacks in wireless networks using signalprints. In *Proceedings of the 5th ACM workshop on Wireless security* (pp. 43–52).: ACM New York, NY, USA.

[Franklin et al., 2006] Franklin, J., McCoy, D., Tabriz, P., Neagoe, V., Van Randwyk, J., & Sicker, D. (2006). Passive data link layer 802.11 wireless device driver fingerprinting. In *In Proceedings of the 15th USENIX Security Symposium* Vancouver, Canada.

[Gast & Loukides, 2002] Gast, M. & Loukides, M. (2002). *802.11 wireless networks: the definitive guide*. O'Reilly & Associates, Inc. Sebastopol, CA, USA.

[Gerdes et al., 2006] Gerdes, R., Daniels, T., Mina, M., & Russell, S. (2006). Device identification via analog signal fingerprinting: A matched filter approach. In *Proceedings of the Network and Distributed System Security Symposium Conference (NDSS 2006)*.

[Guo & Chiueh, 2006] Guo, F. & Chiueh, T. (2006). Sequence number-based MAC address spoof detection. *Lecture Notes in Computer Science*, 3858, 309.

[Hall, 2004] Hall, J. (2004). Enhancing intrusion detection in wireless networks using radio frequency fingerprinting. In *Proceedings of the 3rd IASTED International Conference on Communications, Internet and Information Technology (CIIT)* (pp. 201–206).: Citeseer.

[Hall, 2006] Hall, J. (2006). *Detection of rogue devices in wireless networks*. PhD thesis, Citeseer.

[Hall et al., 2003] Hall, J., Barbeau, M., & Kranakis, E. (2003). Detection of transient in radio frequency fingerprinting using signal phase. *Wireless and Optical Communications*.

[Hall et al., 2005] Hall, J., Barbeau, M., & Kranakis, E. (2005). Radio frequency fingerprinting for intrusion detection in wirless networks. *IEEE Transactions on Defendable and Secure Computing*.

[Hippenstiel & Payal, 1996] Hippenstiel, R. & Payal, Y. (1996). Wavelet based transmitter identification. In *In International Symposium on Signal Processing and its Applications* Gold Coast Australia.

[Hu et al.,] Hu, Y., Perrig, A., & Johnson, D. Packet leashes: a defense against wormhole attacks in wireless networks. In *IEEE INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3.

[IEEE Standards Ass.,] IEEE Standards Ass. *IEEE Standard 802.11a*, http://standards.ieee.org/getieee802/download/802.11a-1999.pdf edition.

[IEEE Standards Ass., 1999] IEEE Standards Ass. (1999). *IEEE Standard 802.11b*, http://standards.ieee.org/getieee802/download/802.11b-1999.pdf edition.

[Kaplan & Stanhope, 1999] Kaplan, D. & Stanhope, D. (1999). Waveform collection for use in wireless telephone identification. US Patent 5,999,806.

[Kasdin, 1995] Kasdin, N. (1995). Discrete simulation of colored noise and stochastic processes and1/f α power law noise generation. *Proceedings of the IEEE*, 83(5), 802–827.

[Kohno et al., 2005] Kohno, T., Broido, A., & Claffy, K. C. (2005). Remote physical device fingerprinting. In *IEEE Symposium Security and Privacy* Washington, DC, USA.

[Langley, 1993] Langley, L. (1993). Specific emitter identification (SEI) and classical parameterfusion technology. In *WESCON/'93. Conference Record,* (pp. 377–381).

[Li & Trappe, 2007] Li, Q. & Trappe, W. (2007). Detecting Spoofing and Anomalous Traffic in Wireless Networks via Forge-Resistant Relationships. *IEEE Transactions on Information Forensics and Security*, 2(4), 793–808.

[Lim et al., 2003] Lim, Y., Schmoyer, T., Levine, J., & Owen, H. (2003). Wireless intrusion detection and response. In *Proceedings of the 2003 IEEE Workshop on Information Assurance*, volume 75: New York: United States Military Academy, West Point.

[Lin, 2000] Lin, L. (2000). *Design Techniques for High Performance Intgrated Frequency Synthesizers for Multi-standard Wireless Communication Applications.* PhD thesis, Citeseer.

[Lyon,] Lyon, G. *Nmap network mapper*, http://nmap.org edition.

[Mathworks,] Mathworks. *Communications Blockset - Phase Noise*, http://www.mathworks.com/access/helpdesk/help/toolbox/commblks edition.

[Mini-Circuits,] Mini-Circuits. *VCO Phase Noise*, www.minicircuits.com/pages/pdfs/vco15-6.pdf edition.

[Mitchell,] Mitchell, T. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill.*

[Montoies, 2008] Montoies, G. (2008). Design of a low-power cmos vco and frequency dividers for a 2.4-ghz bfsk transmitter using an on-chip antenna. Master's thesis, Illinois Institute of Technology.

[Moon et al., 1999] Moon, S., Skelly, P., & Towsley, D. (1999). Estimation and removal of clock skew from network delaymeasurements. In *IEEE INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, volume 1.

[Newsome et al., 2004] Newsome, J., Shi, E., Song, D., & Perrig, A. (2004). The sybil attack in sensor networks: analysis & defenses. In *Proceedings of the 3rd international symposium on Information processing in sensor networks* (pp. 259–268).: ACM New York, NY, USA.

[Packard, 1998] Packard, H. . (1998). Phase noise measurement seminar. In *Literature No. 5968-3734EE*.

[Parno et al., 2005] Parno, B., Perrig, A., & Gligor, V. (2005). Distributed detection of node replication attacks in sensor networks. In *2005 IEEE Symposium on Security and Privacy* (pp. 49–63).

[Patrick, 1991] Patrick, E. (1991). A generalized k-nearest neighbor rule. *Nearest neighbor (NN) norms: nn pattern classification techniques*, (pp.64̃).

[Patwari & Kasera, 2007] Patwari, N. & Kasera, S. (2007). Robust location distinction using temporal link signatures. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking* (pp. 111–122).: ACM New York, NY, USA.

[Paxson, 1998] Paxson, V. (1998). On calibrating measurements of packet transit times. *ACM SIGMETRICS Performance Evaluation Review*, 26(1), 11–21.

[Rasmussen, 2007] Rasmussen, K. B. (2007). Implications of Radio Fingerprinting on the Security of Sensor Networks. *Proceedings of IEEE SecureComm*.

[Razavi, 1996] Razavi, B. (1996). A study of phase noise in CMOS oscillators. *IEEE Journal of Solid-State Circuits*, 31(3), 331–343.

[Remley et al., 2005] Remley, K., Grosvenor, C., Johnk, R., Novotny, D., Hale, P., McKinley, M., Karygiannis, A., & Antonakakis, E. (2005). Electromagnetic signatures of WLAN cards and network security. In *Proc. of the Fifth IEEE International Symposium on Signal Processing and Information Technology* (pp. 484–488).

[Riezenman, 2000] Riezenman, M. (2000). Cellular security: better, but foes still lurk. *IEEE Spectrum*, 37(6), 39–42.

[Shaw & Kinsner, 1989] Shaw, D. & Kinsner, W. (1989). Multifractal modelling of radio transmitter transients for classification. In *Conference Proceedings* (pp. 306).: The Institute.

[Skyworks,] Skyworks. *MICROWAVE PRODUCT DIGEST*, tomdonham.com/new/images/portfolio/advertising/pdfs/ad5.pdf edition.

[Stepanek & Kilpatrick,] Stepanek, P. & Kilpatrick, W. Modeling Uncertainties in a Measuring Receiver. *Agilent Technologies*.

[Talbot et al., 2003] Talbot, K., Duley, P., & Hyatt, M. (2003). Specific Emitter Identification and Verification. *Technology Review*, (pp. 113).

[Tekbas et al., 2004] Tekbas, O., Ureten, O., & Serinken, N. (2004). Improvement of transmitter identification system for low SNR transients. *Electronics Letters*, 40(3), 182–183.

[Ureten & Serinken, 1999] Ureten, O. & Serinken, N. (1999). Detection of radio transmitter turn-on transients. *Electronics Letters*, 35(23), 1996–1997.

[Ureten & Serinken, 2007] Ureten, O. & Serinken, N. (2007). Wireless security through RF fingerprinting. *Electrical and Computer Engineering, Canadian Journal of*, 32(1), 27–33.

[Walker et al., 2000] Walker, J. et al. (2000). Unsafe at any key size; an analysis of the WEP encapsulation. *IEEE document*, 802, 11–00.

[Wang et al., 2005] Wang, B., Omatu, S., & Abe, T. (2005). Identification of the defective transmission devices using the wavelet transform. *IEEE transactions on pattern analysis and machine intelligence*, (pp. 919–928).

[Wikipedia, a] Wikipedia. Discrete wavelet transform.

[Wikipedia, b] Wikipedia. *Manhattan Distance*, http://en.wikipedia.org/wiki/taxicab_geometry edition.

[Wikipedia, c] Wikipedia. *IEEE 802.11*, http://en.wikipedia.org/wiki/ieee_802.11 edition.

[Wikipedia, d] Wikipedia. *IEEE 802.11n*, http://en.wikipedia.org/wiki/ieee_802.11n edition.

[Wikipedia, e] Wikipedia. *k-nearest neighbor algorithm*, http://en.wikipedia.org/wiki/k-nearest_neighbor_algorithm edition.

[Wikipedia, f] Wikipedia. *Phase Noise*, http://en.wikipedia.org/wiki/phase_noise edition.

[Wright, 2003] Wright, J. (2003). Detecting wireless LAN MAC address spoofing. *White Paper, January*.

[Xiao et al., 2007] Xiao, L., Greenstein, L., Mandayam, N., & Trappe, W. (2007). Fingerprints in the ether: Using the physical layer for wireless authentication. In *IEEE International Conference on Communications, 2007. ICC'07* (pp. 4646–4651).

[Zanchi et al., 2001a] Zanchi, A., Samori, C., Lacaita, A., & Levantino, S. (2001a). Impact of AAC design on phase noise performance of VCOs. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 48(6), 537–547.

[Zanchi et al., 2001b] Zanchi, A., Samori, C., Levantino, S., Lacaita, A., & e Inf, D. (2001b). A 2-V 2.5-GHz-104-dBc/Hz at 100 kHz fully integrated VCO withwide-band low-noise automatic amplitude control loop. *IEEE Journal of Solid-State Circuits*, 36(4), 611–619.