Rogue Access Point Detection through Statistical Analysis

Swati Kanaujia

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

Master of Science
in
Computer Engineering

Jung-Min Park, Chair
Yaling Yang
R. Michael Buehrer

May 05th, 2010
Blacksburg, Virginia

Keywords: IEEE 802.11, Rogue Access Point, Intrusion Detection, Hypothesis Test, Naïve Bayes Classifiers

Rogue Access Point Detection through Statistical Analysis

Swati Kanaujia

ABSTRACT

The IEEE 802.11 based Wireless LAN (WLAN) has become increasingly ubiquitous in recent years. However, due to the broadcast nature of wireless communication, attackers can exploit the existing vulnerabilities in IEEE 802.11 to launch various types of attacks in wireless and wired networks.

This thesis presents a statistical based hybrid Intrusion Detection System (IDS) for Rogue Access Point (RAP) detection, which employs distributed monitoring devices to monitor on 802.11 link layer activities and a centralized detection module at a gateway router to achieve higher accuracy in detection of rogue devices. This detection approach is scalable, non-intrusive and does not require any specialized hardware. It is designed to utilize the existing wireless LAN infrastructure and is independent of 802.11a/b/g/n. It works on passive monitoring of wired and wireless traffic, and hence is easy to manage and maintain. In addition, this approach requires monitoring a smaller number of packets for detection as compared to other detection approaches in a heterogeneous network comprised of wireless and wired subnets.

Centralized detection is done at a gateway router by differentiating wired and wireless TCP traffic using Weighted Sequential Hypothesis Testing on inter-arrival time of TCP ACK-pairs. A decentralized module takes care of detection of MAC spoofing and totally relies on 802.11 beacon frames. Detection is done through analysis of the clock skew and the Received Signal Strength (RSS) as fingerprints using a naïve Bayes classifier to detect presence of rogue APs.

Analysis of the system and extensive experiments in various scenarios on a real system have proven the efficiency and accuracy of the approach with few false positives/negatives and low computational and storage overhead.

This thesis is dedicated to my beloved mother, Late Mrs. Archana Kanaujia,

and to

my parents, siblings and dearest husband

for their constant support, love and encouragement

Acknowledgments

I would like to express gratitude to my parents, Mr. Ramesh Chandra Kanaujia and Mrs. Seema Kanaujia, siblings – Sakshi, Akash and Shreyash who have always had great confidence in me. I want to thank Rishi, my husband, for continuously motivating me and being immensely supportive. Your unquestioned faith and selfless love instilled confidence in me to surmount every hurdles of my life. I also thank to my in-laws, brother-in-law and sister-in-law for their support and love.

I thank my advisor Dr. Jung-Min Park for his invaluable support, guidance and giving me opportunity to work under him. It has been a great learning experience working with him for last two years. The constructive suggestions and criticisms offered by him have been very helpful in shaping up this research work. I would like to thank my committee members Dr. Yaling Yang and Dr. R. Michael Buehrer for their valuable advice over the past year. A special thanks to Dr. Inyoung Kim and Dr. Lynn Abbott as their suggestions, comments and additional guidance were invaluable to the completion of this work.

I thank all the members of Arias Lab at VT for their support. Their feedback and suggestions from time to time which helped me to improve the quality of this thesis. In particular, I would like to thank Amol A. Deshpande, Jatin S. Thakkar and Kaigui Bian with whom I had many informal discussions about my research which helped me to expedite my work. I am thankful to Mr. Randy Marchany for using his lab for testbed setup. Special thanks to John Paul to let me use his basement for a interference free setup for my wireless data collection. I also thank Nikhil Rahagude for proof reading my manuscript.

This section would not complete without the mention of my close friends who always helped me to be a better person, especially; Rakshita, Archana, Emma, Prashant, Alok, Siddhartha, Akshat, Somya, Gayatri, Shraddha and Shivanand. I would like to thank you all for your love, support, meaningful conversations, fun and good laughs. Also, I thank all my friends who have made my stay in Blacksburg very enjoyable.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Various types of wireless networks have gained significant popularity in recent years. Their attractiveness have transitioned from academic and specialized military applications to day-to-day businesses. Nowadays it provides wireless access to the offices, homes, public hot-spots and other locations. One of the most widely deployed wireless networks is the *IEEE 802.11 based Wireless Local Area Networks (WLANs)* [1], also known as *Wi-Fi*. The IEEE 802.11 standards come in various flavors and the most widely adopted ones are a/b/g/n. 802.11b provides a maximum data rata of 11 Mbps and operates at 2.4 GHz. Similarly, 802.11g also operates at 2.4 GHz but provides faster connectivity of 54 Mbps using Orthogonal Frequency Division Multiplexing (OFDM). 802.11a also uses OFDM with maximum throughput of 54 Mpbs but operates at the 5 GHz band. 802.11n is a recent addition to the family and uses Multiple-Input Multiple-Output (MIMO) and many other advanced features and achieves much higher throughput in the range of hundreds of Mbps. These standards can be used in conjunction with 802.11e for quality of service and 802.11i for security.

Advantageously, WLANs facilitate improvement in productivity through mobility, provide a quick and effective way of wireless extension to an existing LAN, and gives access to places where wired connectivity is impossible or too expensive to install. In order to provide extension, one or more *Access Points* (APs) can connect to the connection ports either

directly or through hubs/switches. A user can access the LAN using a device known as a *station* which has a Wi-Fi radio. The station can wirelessly communicate with the AP.

Generally, to connect to a wired LAN, a physical access is required through a network port. However, to connect to wireless LANs, a user only needs to be in the coverage range of an AP. Security of the wired network has focused on controlling access to the physical space where the LAN connection ports are located by using management applications such as firewalls, virus scanners and Intrusion Detection Systems (IDSs). Wireless LANs use radio waves which cannot be contained in the physical space bounded by physical structures and pass through modern building materials. Hence, wireless LAN coverage is not limited to the inside of building walls. Thus, wireless communications can introduce additional security threats. The radio waves appear over the air and hence transmissions from Wireless LANs can be monitored by *unauthorized users* with suitable equipments. Access to a corporate network can be achieved from outside a building using readily available technology. Consequently, one cannot ensure security for a wireless LAN by controlling physical access to the connection ports.

The security mechanism provided by the base 802.11 standards suffers from a number of fundamental flaws and can easily be circumvented or evaded. To prevent unauthorized access to the wireless network, the AP can employ certain techniques. For example, in accordance with 802.11, a user currently requests to carry out an authentication handshake with the AP before being able to connect to the network. Examples of such handshakes are Wireless Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA[1] and WPA2 [2]) based shared key authentication, 802.1x based port access control, and 802.11i based authentication. Until ratification of 802.11i, Layer 3 security mechanisms such as *Virtual Private Networks (VPNs)* were used to secure WLAN access. The AP can provide additional security measures such as encryption and firewalls. This thesis discusses existing security vulnerabilities in IEEE

[1]Protocol proposed in response to several serious weaknesses researchers had found in WEP.

[2]WPA2 has replaced WPA; WPA2 requires testing and certification by the Wi-Fi Alliance. WPA2 implements the mandatory elements of 802.11i.

802.11 and proposes a countermeasure for one of the security threats.

Section 1.1 discusses the motivation behind the research work. The research aims and contributions are summarized in Section 1.2. Finally the thesis organization is presented in Section 1.3.

1.1 Motivation

Despite the above-mentioned enhanced security measures, security vulnerabilities still persist and wireless networks remain vulnerable to a number of attacks. In the past, research on the security of wireless networks centered on securing APs from unauthorized malicious clients, since APs were deemed vulnerable and exposed to malicious entities. Now, there has been a shift in the focus of attention towards protecting clients in wireless networks and an important threat in this respect is from rogue APs. For example, an *unauthorized or Rogue AP (RAP)* may connect to the LAN and then, in turn, allow unauthorized users to connect to the LAN. These unauthorized users can thereby access proprietary/trade secret information on computer systems connected to the LAN without the knowledge of the owner of the LAN. Notably, an unauthorized AP can easily masquerade as an authorized AP. That is, a rogue AP can advertise the same feature set (e.g., MAC address and other settings, which is known as *MAC Spoofing*) as an authorized/legitimate AP, thereby making its detection difficult. From a security perspective, this scenario is extremely dangerous as it creates a "backdoor" opening for malicious parties to gain access to the network. The danger is compounded by the fact that system administrators may be totally unaware that the vulnerability even exists. This type of rogue APs poses a serious security threat due to its ability to hide within existing networks and its potential for supporting mischievous activities.

Moreover, even if a rogue AP is not LAN-connected, it may still pose a security threat. Specifically, authorized clients in communication with the rogue AP may be unwittingly providing proprietary/trade secret information to the rogue AP. Even an employee with

basic computer skills can purchase an inexpensive AP and quickly configure/install it into a corporate network. Rogue APs pose serious security threats to secured networks. They can steal sensitive information or even launch attacks to the network. Furthermore, rogue APs may interfere with nearby authorized APs and lead to performance problems inside the network. Therefore, a need arises for a security solution that ensures security for wireless LANs. A typical scenario in which a rogue AP can pose a security threat is shown in Figure 1.1.

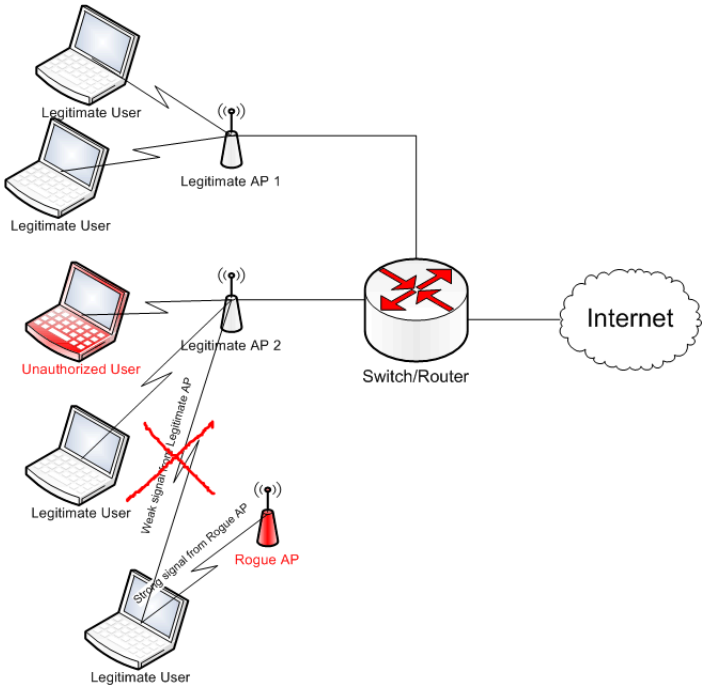


Figure 1.1: Rogue Access Point and Rogue Client in Network

Unfortunately, regardless of the ratification of the IEEE 802.11 standards, 802.11 WLANs still suffer from a number of security vulnerabilities. WPA has resolved a number of security issues, like encrypting 802.11 data frames to make them resistant to spoofing and replay attacks. However, some vulnerabilities still persist e.g., unprotected frames in IEEE 802.11 such as Management, Control and Extensible Authentication Protocol (EAP) frames. Furthermore, even if authentication support is available in next-generation WLAN equipment, it cannot protect the already deployed legacy WLAN devices. These unprotected frames

can easily be forged and used to launch attacks on the wireless networks. This problem is further exacerbated by the fact that almost all WLAN hardware permits the user to change its MAC address, and hence WLANs remain susceptible to identity attacks and denial–of–service attacks even with WPA enabled.

Currently, detection of rogue APs depends on very rudimental methods. One reasonable method for detection is to search for signals or beacons of rogue APs. The motivation of this research is to develop novel lightweight hybrid wireless intrusion detection techniques that are capable of detecting rogue APs by performing statistical analysis of wired and wireless traffic. Towards this objective, we propose a suite of lightweight security solutions for wireless networks that complements the use of conventional authentication services and is capable of detecting multiple devices using the same network identity.

1.2 Contributions

This detection approach is scalable and non-intrusive, does not require any specialized hardware, is designed to utilize the existing wireless LAN infrastructure and independent of the differences between the wireless LAN standards, viz 802.11a/b/g/n. It works on passive monitoring of wired and wireless traffic and hence is easy to manage and maintain. The primary contributions of this research are summarized below:

1. Development of an algorithm based on weighted sequential hypothesis test applied to packet-header data that are passively collected at a monitoring point. The algorithm exploits the fundamental properties of the 802.11 CSMA/CA mechanism and the half duplex nature of wireless channels to differentiate wired and wireless LAN TCP traffic by taking into account the *Operating System (OS) diversity (specifically Windows)*. Once TCP ACK-pairs are observed, prompt decisions are made with little computational and storage overhead.[3]

[3]This work is an extension of [2].

2. The main contribution of this research is a new wireless fingerprinting scheme that differentiates between unique devices through timing analysis of 802.11 beacon frames. More specifically, a unique device is defined as a unique combination of clock skew[4] and RSS. This fingerprinting technique distinguishes between different combinations of these two features. In contrast to existing techniques, this approach is based on the timing analysis of 802.11 beacon frames, which is passive and non-invasive.

1.3 Thesis Organization

This thesis has been divided into five major chapters, which are structured around the motivation of the research (as detailed in Section 1.1).

Chapter 2 provides background material regarding IEEE 802.11 WLANs and their operation and also discusses about security vulnerabilities which exist in current standard versions. It also provides a discussion on current types of rogue APs which exist in wireless networks and techniques for wireless detection which can be broadly classified into over the air (wireless monitoring) and on the wire (wired monitoring). A comprehensive review of the research currently going on, in the field of detection for rogue APs wraps up this chapter.

Chapter 3 proposes an algorithm to detect rogue AP by differentiating wired and wireless traffic by applying weighted sequential hypothesis test on TCP ACK-pairs. It also discusses the role of operating system in this analysis and how it effects the inter-ACK time between two TCP ACK. Then it gives details of the detector module which takes care of the operating system diversity.

Chapter 4 proposes a novel classification based intrusion detection scheme which utilizes device fingerprinting. It discusses about how clock skew and revoiced signal strength can be used to fingerprint a device and the effectiveness of this scheme is measured through *naïve*

[4]Clock skews are the inherent tiny drifts in the clocks of hardware devices due to variations in the manufacturing process.

Bayes classifier.

Chapter 5 discusses the testbeds which are used in various analysis and experiments those are conducted using different scenarios based on wired and wireless side detection schemes. The detection results from each detection technique are then analyzed to test the expected true positive and false positive rates per device for each detection technique.

Chapter 6 summarizes the research work. It presents several concluding remarks and describes future directions for research.

Chapter 2

IEEE 802.11 MAC and Related Work

A wireless local area network (WLAN) links devices via a wireless distribution method (typically spread-spectrum or OFDM radio), and usually provides a connection through an access point to the wider internet. Currently, most wireless networks (WLANs) are based on the IEEE 802.11b, 802.11a, 802.11g or 802.11n standards which define how to wirelessly connect computers or devices to a network. This gives users the flexibility and mobility to move around within a local coverage area and still be connected to the network. WLANs are now becoming a viable alternative to traditional wired solutions. For example, hospitals, universities, airports, hotels and retail shops are already using wireless technologies to conduct their daily business operations; sometimes for free.

It is well-known that due to the broadcast nature of the wireless access it is open to many malicious attacks. Access to the physical medium is restricted by cables and buildings; however no such restrictions apply to the wireless medium. The broadcast nature of the medium exposes WLANs to passive eavesdropping.

The WLAN standards acknowledge the security threats to WLANs and provide link layer security mechanisms to address them. However, early attempts at such mechanisms failed to address the security threats effectively and the WLAN security mechanisms have had to go

through a number of revisions to get to their current robust and reliable state. Unfortunately, they still suffer from a number of vulnerabilities that can potentially be exploited to launch a number of attacks against WLANs.

This chapter provides the relevant background on WLAN operations and reviews the evolution of WLAN security needed to understand the framework which is used in subsequent chapters 3 and 4. It also reviews all outstanding issues and attacks that can be used against protected WLANs by the latest WLAN security mechanisms. The IEEE 802.11 architecture and operations are discussed in Section 2.1. A review of security vulnerabilities and existing threats to 802.11 WLAN are presented in Section 2.2. Section 2.3 discusses about taxonomy of the Rogue APs and Section 2.4 provides the overview of the current monitoring methodologies to detect the anomaly in WLANs. These discussions are followed by review of the current research going on in field of security of the WLANs in Section 2.5.

2.1 IEEE 802.11 Basics

The primary goal of the standard was the specification of a simple and robust WLAN which offers time-bounded and asynchronous MAC Service Data Unit (MSDU) delivery services. The MAC layer should be able to operate with multiple physical layers, each of which exhibits a different medium sense and transmission characteristics [1] [3]. Additional features of the WLAN include the support of power management to save battery power; the handling of hidden nodes [1]; description of the requirements and procedures to provide data confidentiality of user information being transferred over the wireless medium (WM) and authentication; and the ability to operate worldwide. The following section will introduce the system and protocol architecture of the IEEE 802.11.

[1] Node refer to any entity that is capable of communicating using 802.11

2.1.1 IEEE 802.11 Architecture

The wireless networks can be divided into two structural configurations or designs as follows:

2.1.1.1 Infrastructure Architecture

Figure 2.1 shows the components of an infrastructure architecture [3]. Several nodes, called *stations (STAs)* are connected to *Access Points (APs)*. Stations are terminals with access mechanisms to the wireless medium and radio contact to the AP. The stations and the AP which are within the same radio coverage form a *Basic Service Set (BSS)*. In the Figure 2.1 two BSSs which are connected via a *Distribution System (DS)*. *Extended Service Set (ESS)* is a set of one or more interconnected BSSs that appears as a single BSS to the logical link control (LLC) layer at any STA associated with one of those BSSs. The DS connects the wireless networks via the APs with a *portal*, which forms the interworking unit to other LANs.

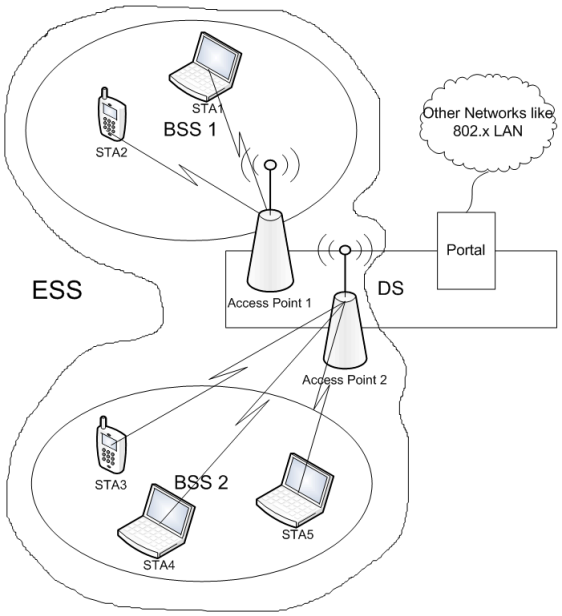


Figure 2.1: Architecture of an Infrastructure-based IEEE 802.11

2.1.1.2 Ad-Hoc or Independent BSS Architecture

An *ad-hoc* or *Independent BSS (IBSS)* has no central control entity such as an AP but is comprised of several STAs in direct communication with each other over the wireless medium. STAs in an IBSS communicate directly with each other and hence must be within direct radio communication range and use the same radio frequency. One of the key advantages of ad-hoc WLANs is that theoretically they can be formed any time and anywhere, allowing multiple users to create wireless connections cheaply, quickly and easily with minimal hardware and user maintenance.

This thesis focuses entirely on infrastructure based architecture and hereafter, all references to WLANs refer to infrastructure BSSs.

2.1.2 IEEE 802.11 Medium Access Mechanisms

IEEE 802.11 uses a *Carrier Sense Multiple Access (CSMA)* scheme to control access to the wireless transmission medium. It is a random access scheme with carrier sense and collision avoidance through *random backoff*. The medium can be busy or idle which is detected by the *Contention Channel Assessment (CCA)*. This mechanism is shown is Figure 2.2 [1] and summarized as *Distributed Coordination Function (DCF)* in [1]. If the medium is idle for at least the duration of *DCF Inter-Frame Space (DIFS)*, a STA can access the medium at once. But if medium is busy, STA has to wait for the duration of DIFS, entering a contention phase afterwards. Each STA can choose a random backoff time within a *contention window* and delays medium access for this random amount of time. STA decrements the backoff timer while medium is idle and can access the medium when timer is decremented to zero. A successful transmission is evaluated by a positive ACK which is send by the receiver to sender after receiving the frame. A refinement of the method may be used under various circumstances to further minimize collisions by the transmitting and receiving STA exchange short control frames (RTS and CTS frames). These frames are exchanged after determining

that the medium is idle and prior to data transmission. This method is known as *virtual carrier sensing*. For further discussion, detail mechanism is covered in [1].

Another optional access method is *Point Coordination Function (PCF)*, which requires central entity to control the nodes within BSS and hence can only be used for infrastructure networks. PCF is not discussed any further in this thesis.

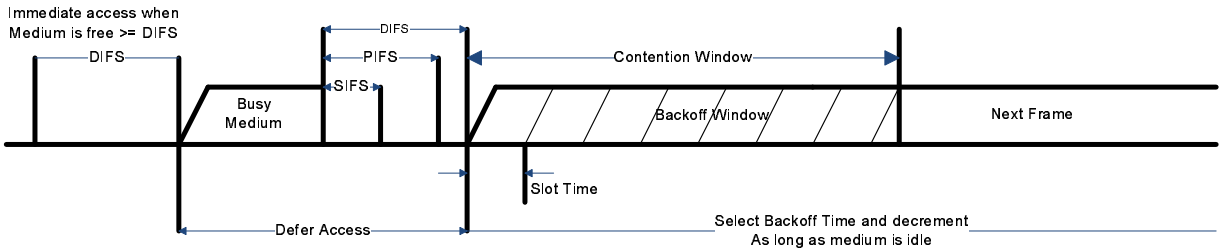


Figure 2.2: IEEE 802.11 Medium Access

2.1.3 IEEE 802.11 MAC Frame Format

Each MAC frame consists of a MAC header, a variable length frame body (maximum size of 2304 octets) and FCS [1]. Figure 2.3 depicts the general MAC frame format. The first three fields (frame control, duration/ID, and address 1) and the last field (FCS) in Figure 2.3 constitute the minimal frame format and are present in all frames. The fields address 2, address 3, sequence control, address 4, QoS control and frame body are present only in certain frame types and subtypes.

The frame control field consists of the following subfields: protocol version, type, subtype, to DS, from DS, more fragments, retry, power management, more data, protected frame and order. The format of the frame control field is illustrated in Figure 2.3.

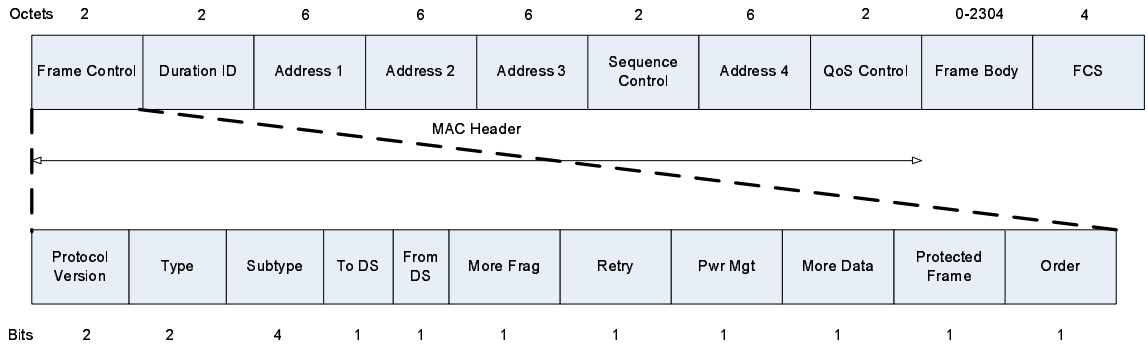


Figure 2.3: IEEE 802.11 MAC frame format

There are three types of frame in 802.11 WLANs: *control frames, management frames* and *data frames*. Each type of frame has its own subtype as shown in Table 2.1. Details can be found in [1]. Management frames are mainly used for network management and admission control. Control frames are mainly used for access control, and data frames carry data.

Table 2.1: IEEE 802.11 Frame Subtypes

Control frame subtypes	Management frame subtypes	Data frame subtypes
PS-Poll (Power Save Poll)	Association Request	Data
RTS (Request To Send)	Association Response	Data + CF-ACK
CTS (Clear To Send)	Reassociation Request	Data + CF-Poll
ACK (Acknowledgement)	Reassociation Response	Data + CF-ACK + CF-Poll
CF-End (Contention Free End)	Probe Request	Null Function (no data)
CF-End + CF-ACK	Probe Response	CF-ACK (no data)
	Beacon	CF-Poll (no data)
	ATIM (Ad Hoc Traffic Indication Map)	QoS Data
	Disassociation	QoS Data + CF-ACK
	Authentication	QoS Data + CF-Poll
	Deauthentication	QoS Data + CF-ACK + CF-Poll
	Action	Null QoS Data
	Block ACK Request	Null QoS Data + CF-Poll
	Block ACK	Null QoS Data + CF-ACK + CF-Poll

2.1.4 IEEE 802.11 Operations

All APs advertise their presence by transmitting *beacon frames* at regular intervals. STAs listen for beacon frames to identify the APs within range and synchronize their timer with the timer in the AP, the STA is associated with, via a *Timer Synchronization Function (TSF)*. There are many different *channels* that an STA can use and tune into each channel and in turn listen beacon frames in each channel. This process is called *scanning*. Alternatively, *probe requests* frames are sent by STAs actively searching for APs, known as *probing*. STAs are identified through their MAC address and AP responses back by sending *probe response* frame, which is similar to beacon frame and contains necessary information of the BSS. After discovering the APs, STA can choose whom to connect based on signal strength. When

STA is ready to connect, it sends *authentication request* and in exchange, *authentication response* is sent by AP. A station can be authenticated by multiple APs; however it should be associated with only one AP at a time. After the authentication, *association request* and *association response* frames are exchanged to establish the association. *Deauthentication* frame is used to return to initial state and *deassociation* frame is used for disassociation.

2.2 Security Vulnerabilities of IEEE 802.11

Although wireless communication provides a lot benefits to users such as mobility, flexibility and scalability, even then it has its own inherent risks, some of them are similar to wired networks and some are due to wireless medium. These risks are because of shared nature of the wireless medium which opens the network to everyone. Even with tools such as encryption being available, this technology is still vulnerable to attacks and can be compromised. This section provides an overview of IEEE 802.11 security. It begins by explaining the main security concerns and threats against WLANs. Next, it reviews the security features and weaknesses of IEEE 802.11.

2.2.1 WLAN Security Concerns

The security objectives of WLANs are similar to those of wired LANs and other wireless networks and they are as follows [4] [5]:

- **Confidentiality**–No unauthorized parties should able to read the communication between two authorized nodes.
- **Integrity**–Integrity guarantees that the content of a communication has not been altered in transit.
- **Availability**–The WLAN and its resources should be accessible to all individuals and

devices on demand.

- **Access Control**–The WLAN should restrict the rights of the devices and individuals to access the network and its resources.

2.2.2 WLAN Security Threats

Although there has been tremendous growth and success, everything relative to 802.11 WLANs has not been positive. There have been numerous published reports and papers describing attacks on 802.11 wireless networks that expose organizations to security attacks. WLAN threats typically involve an attacker with access to the radio link between two STAs or between STA and an AP. The most significant difference between protecting wireless and wired LANs is the relative ease of intercepting and injecting network communications. The nature of the attackers classifies the attacks. In *passive attacks*, the attacker can only monitor the traffic and analyze it or eavesdrop. But in *active attacks*, the attacker can also alter the information or obstruct it. Depending on attacker threats, WLANs attacks are divided into following categories [4] [5]:

- **Eavesdropping**–Attacker can passively monitor and capture the traffic between two unauthorized parties. Later on, he can interpret it.
- **Traffic Analysis**–It is the process of intercepting and examining messages in order to deduce information from patterns in communication. It can be performed even when the messages are encrypted.
- **Masquerading**–An adversary assumes the identity of one of the legitimate parties in a network and uses it to eavesdrop.
- **Replay**–In this attack an adversary monitors the ongoing session and replays the entire session later as authorized user.

- **Message Modification**–Data being transmitted between two communicating nodes is deleted, added, changed or reordered by an adversary.
- **Message Injection**–In this attack the adversary inserts data claiming that it is from an legitimate source.
- **Man-in-the-Middle**–Adversary can intercept the ongoing session by monitoring unprotected frames and then impersonates as legitimate user and modifies contents in between.
- **Denial-of-service**–A user or organization is deprived of the services of a management resource they would normally expect to have.
- **Session Hijacking**–Attacker sniffs ongoing session between AP and STA. Then attacker causes STA to drop its connection right after it passes the authentication by spoofed AP by sending a forged de-authentication frame. After that the attacker impersonates the original wireless station to AP.

2.2.3 WLAN Security Mechanism

In IEEE 802.11, two types of authentication are provided. First one is *open system authentication*, which is compulsory in 802.11, is effectively a misnomer and does not provide true identity verification and STA is authenticated by *Service Set Identifier (SSID)*, assigned name to WLAN and 48-bit *MAC address* of STA. These two parameters can be easily spoofed and attackers can gain unauthorized access easily. Second one is *shared key authentication*, which uses challenge-response scheme, but due to weakness in it, it is susceptible to eavesdropping and man-in-middle attack. Both of these authentication mechanisms do not require mutual authentication and hence rogue AP can be easily installed. In following subsections, a brief discussion of mechanisms is given, which is provided by IEEE 802.11:

2.2.3.1 Wired Equivalent Privacy (WEP)

WEP protocol was first introduced to protect confidentiality in 802.11 WLAN by using Rivest Cipher 4 (RC4). It encrypts data frames by protecting their contents from disclosure to eavesdroppers. It also supports 40-bit WEP key, even 128 and 256 bits keys are also introduced. However, WEP suffers from many flaws as it is susceptible to brute force attacks and it does not provide any protection against replay attacks due to lack of incrementing counter. For authentication, WEP uses shared key authentication, STA encrypts the challenge using WEP key and returns the result to AP. As discussed above, it is not a mutual authentication as the AP is not authenticated by the STA, which opens the door for a rogue AP. WEP also suffers from cryptographic weakness as RC4 employs 24-bit initialization vector (IV), which is too small to prevent recurring IVs on a busy WLAN. In addition to that, non-cryptographic checksum is used for data integrity and an adversary can alter both data and the corresponding checksums without detection. Hence WEP establishes a false sense of security that causes people to be more willing to send sensitive data over the network.

2.2.3.2 WiFi Protected Access (WPA)

While examining the shortcomings of IEEE 802.11 WEP and starting to develop the 802.11i amendment, *Wi-Fi Protected Access (WPA)*, was promoted by the WiFi Alliance[2]. WPA is an interim security solution that does not require a hardware upgrade in existing 802.11 equipment. WPA is not a perfect solution but is an attempt to quickly and proactively deliver enhanced protection to address some of the problems with WEP prior to the full-blown security techniques of IEEE 802.11i. WPA is essentially a subset of the draft IEEE 802.11i requirements available at that time and is significantly different from 802.11i as it does not support Advanced Encryption Standard (AES), a strong encryption algorithm. WPA uses *Temporal Key Integrity Protocol (TKIP)* for confidentiality which still uses RC4 but includes an extended IV space and key mixing function to construct per packet keys.

[2] www.wi-fi.org

However, due to legacy hardware limitation WPA uses weak keyed Message Integrity Code (MIC) computed using the Michael algorithm. Authentication is done either through *Pre-Shared Key (PSK)* by using 128 bit encryption key and a 64 bit MIC key or can be done through IEEE 802.1X[3] and EAP. However, due to legacy hardware limitation, WPA also has some weakness as it is possible to find the MIC given one per packet key. This shows that parts of TKIP are weak on their own.

WPA2 extends WPA and requires testing and certification by the Wi-Fi Alliance. WPA2 implements the full set of IEEE 802.11i requirements, however WPA2 is backward compatible with WPA. In particular, it introduces a new AES-based algorithm, *Counter mode with Cipher block chaining Message authentication code Protocol (CCMP)*, which is considered fully secure and requires new hardware. Unlike TKIP, key management and message integrity is handled by a single component built around AES using a 128-bit key, a 128-bit block. Both WPA and WPA2 have two modes of operation: Personal and Enterprise. The personal mode involves the use of a pre-shared key for authentication, while the enterprise mode uses IEEE 802.1X and EAP for this purpose.

2.2.3.3 IEEE 802.11i

IEEE 802.11i was ratified in 2004 and is the sixth amendment to the baseline IEEE 802.11 standard. It is designed to be the long term solution for WLAN security issues. The IEEE 802.11i specification introduces the concept of an *Robust Security Network (RSN)*, which is a wireless network that allows the creation of *Robust Security Network Associations (RSNA)* only. RSNAs are logical connections between communicating IEEE 802.11 entities. They are established through the IEEE 802.11i key management scheme called as *4-Way Handshake* [5]. RSNAs allow for the protection of data frames and provide enhanced security relative to the flawed WEP. The IEEE 802.1X framework specified by the IEEE 802.11i amendment provides the means to block user access until authentication is successful, thereby

[3]The 802.1X port-based access control provides a framework to allow the use of robust upper layer authentication protocols.

controlling access to the WLAN resources. The technique used to block access is known as port-based access control. Confidentiality and integrity is achieved by CCMP and TKIP, whereas TKIP is optional, CCMP is mandatory for RSNA compliance.

However, 802.11i does not mitigate any risk against PHY and MAC based WLAN DoS attacks and hence all management, control frames and EAP frames are unprotected and can be used to launch DoS attacks against the WLANs. Moreover, legacy wireless networks, which are widely deployed cannot be upgraded to 802.11i soon in near future and hence legacy WLAN networks still suffer from the above mentioned security threats. And hence, a need arises to handle security threats which exist in network in conjunction to existing WLAN security mechanisms (WEP and WPA) today. Next section demonstrates the classification of rogue APs which exist in today's WLAN networks.

2.3 Rogue AP Classification

As discussed in sections 1.1 and 2.2 rogue APs are very harmful for wireless network and pose many type of security threats in wireless networks. A comprehensive taxonomy of rogue AP detailing different categories of rogue APs has been presented by Ma *et. al.* [6]. The authors have classified the rogue AP in mainly four categories namely improperly configured AP, unauthorized AP, phishing AP and compromised AP which are defined below:

2.3.1 Improperly Configured AP

There are many common situations in which an authorized AP can become a rogue AP because of some configuration problems or mistakes. The most common scenario is when a network administrator or employee fails to set up the AP properly due to a lack of knowledge of the organization's security policy. For example, in open mode authentication any wireless client device in state when it is unauthenticated and un-associated, can send authentication

requests to an AP. After successful authentication, a client would move to authenticated but un-associated state. If an AP does not validate the client properly due to a configuration flaw, an attacker can send lot of such authentication requests, overflow the AP's client-association-table, and make it to reject access to other clients including the legitimate ones. Other possible reasons are device driver malfunction, faulty software or firmware update or network cards operating in more than one mode together. Furthermore, ad hoc mode of a wireless station without strong security measures may enable attackers to intrude a connected network via this station when it utilizes both wired and wireless interfaces [6].

2.3.2 Unauthorized AP

Employees can buy an AP (which are readily available in market with cheap cost) and install it on the company's LAN for their own convenient network uses without the authorization of network administrator. This type of rogue AP creates large vulnerability to the enterprise network. It enables unauthorized users or attackers from outside to access the company's network. They can retrieve confidential data, send objectionable content to others, attack company assets, or use company's network to attack others. This type of rogue AP is very common especially in the organization that lacks of wireless security policy and security awareness training for employees.

Another category is neighborhood rogue AP, in which the AP is setup by an other company in the close vicinity. The administrator has no authority to control or shut down legitimate APs of the other company. Unintentionally connecting to the neighborhood AP would compromise the security and can expose sensitive data. For ad-hoc devices, wireless clients can communicate among themselves without requiring a LAN bridging device such as an AP. Though such devices can essentially share data among themselves, they pose a significant threat to the enterprise as they lack the necessary security measures such as 802.1x user authentication and the dynamic key encryption. As a result, ad-hoc networks risk exposing data in the air (as data is not encrypted). In addition, weak authentication may allow

unauthorized devices to associate. If the ad-hoc mode clients are also connected to the wired network, the entire enterprise wired network is at risk. [6].

2.3.3 Phishing AP

The rogue AP is setup outside the company by the attacker and does not connect to the company's network. Typically the attacker will use the high transmission power and high antenna gain rogue AP with the spoof credentials like MAC, SSID etc. It aims to allure the target employee to connect the rogue AP. All user traffic is redirected through the rogue AP and analyzed by the attacker. This attack is called man-in-the-middle attack that does not require mutual authentication. These phishing APs are also known as *"evil twins"*, always pretend to be legitimate APs in the enterprise and, thus, they broadcast beacon frames which overhear from legitimate APs. There are number of freely available tools which allow to clone the MAC address and allows spoofing attacks [6].

2.3.4 Compromised AP

WEP can be easily cracked by many open-source attack tools which are available in market. Recently even WPA-PSK is also hacked by hacking softwares/tools. Hence, even a layman can easily breach the security aspect of the enterprise network and can launch the attack through any of legitimate AP in the network. This type of rogue APs are hard to detect because they are considered as legitimate APs by network administrators. Attackers can launch various type of attacks and can gain access to potentially sensitive data of the company [6].

Table 2.2 summarizes the types of rogue APs and a number of possible scenarios [6]:

Table 2.2: Classification of Rogue AP and possible scenarios

Rogue AP Class	Possible Scenarios
Improperly configured	Insufficient security knowledge; faulty driver; physically defective; multiple network cards
Unauthorized	Connected to internal LAN without permission; external neighborhood AP
Phishing	Fabricated by adversary
Compromised	Disclosure of security credentials

2.4 Monitoring Methods

In this section, we will discuss about wireless network traffic monitoring methodologies. Mainly these measurements are conducted for diagnosis; network performance and anomaly detection. There are two common practices for these measurements. One is done at wired network side while other at wireless network side, known as *wired monitoring* and *wireless monitoring* respectively. Although both of them are designed to detect security events of interest, they both operate at different OSI layers. Wired monitoring concentrates on OSI layers 3 (network) and above; the wireless monitoring specializes in detecting attacks exploiting protocols and mechanisms on wireless medium characteristics, such as PHY/MAC of IEEE 802.11. Next two subsections discuss about both methodologies stating their advantages and disadvantages.

2.4.1 Wired Side

In wired monitoring, network traffic (Layer 3 and above) is monitored from choked points like gateway, router etc. It is used for maximum visibility at these points. Data collected is used for further analysis. Another approach is use of network management software to

discover APs. These software use multiple protocols to detect devices connected in the LAN, including Simple Network Management Protocol (SNMP), Telnet, Cisco Discovery Protocol (CDP – specific to Cisco devices) etc. This approach is very reliable and well to proven to detect an AP anywhere in the LAN irrespective of its physical location. For example, in case of SNMP, a management machine is connected to a wired portion of a LAN, machine periodically sends SNMP queries, collects responses from APs and obtains statistics [7]. Since, SNMP is subject to periodical polls of device's state information by a local manager, the results are not sensitive to instantaneous conditional changes. Another limitation with this method is that any AP which does not support SNMP/Telnet etc., will go unnoticed by the network management software.

In the former method, Round Trip Time (RTT) or inter-packet arrival time is used for determining the performance issues like Quality of Service (QoS), optimization etc. on one or more wireless links in the communication path in hybrid wired and wireless environments by determining whether the traffic originated on a wireless link or a wired link. The advantages of wired monitoring are that it is inherently independent of wireless protocol, frequency, signal range, it cannot be easily evaded by attackers, and it can be performed at a central location which makes it cost effective and easy to manage.

However wired monitoring is unable to detect attacks based on 802.11 unprotected frames, malicious payload switched directly between communication station via the AP, PHY and MAC layer based jamming attacks. Wired monitoring does not monitor the WLAN activity for security policy compliance violations and is unable to determine whether a WLAN node is MAC spoofed.

2.4.2 Wireless Side

In wireless monitoring, a number of *sniffers* are deployed in existing wireless networks in order to observe 802.11 (PHY/MAC) traffic characteristics [7] which in turn can be used for wireless performance measurement or for anomaly detection. Generally these sniffers are de-

ployed within the range of AP's radio for accurate and sensitive capturing and are completely independent of the operational network without any interference to existing infrastructure.

Unlike wired monitoring, wireless monitoring is quite sensitive to physical information, such as wireless medium itself. It allows to analyze physical layer header information such as signal strength, noise level and data rate as well as allows to examine link layer headers, which include IEEE 802.11 type and control, management fields [7]. In addition, error rates and throughput performance can be determined from physical layer information. These analysis can be used for site planning. Moreover, from MAC layer data, traffic characterization can be done based on frame type and can help in diagnosing the problems of wireless networks. Wireless monitoring provides assistance in identifying the physical location of an unauthorized node, rogue AP or an adversary by using the distributed nature of their sniffer throughout the WLAN.

However, wireless monitoring poses numerous challenges [7] as sniffers are limited by their disk sizes, processing powers and signal receiving ranges. Furthermore, their location of work effectiveness needs a lot of planning. Even, it is difficult to collect and synchronize a lot of data collected from multiple sniffers. To exacerbate, different 802.11 PHY layers tend to operate at different frequencies and permit the existence of a number of radio communication channels for the WLAN nodes to communicate over. Hence to be able to capture all traffic on the WLAN, dedicated sniffers have to be deployed for monitoring each channel and frequency throughout the WLAN or the sniffers have to use some sort of sampling algorithm where the sensor periodically switches between different channels and frequencies. The biggest challenge is the scalability and cost which can be significant for a large number of monitoring systems. Additionally, if enterprise network covers a large number of geographically dispersed locations, this method of rogue detection may be unworkable.

2.5 Related Work

This section provides a review of all the current research in the field of rogue AP that are commercially available or have been academically published. As mentioned in Section 2.4, monitoring IEEE 802.11 RF waves and IP traffic are two broad classes of approaches to detecting rogue APs.

2.5.1 Commercial Products

There are currently a number of Intrusion Detection Systems (IDSs) on the market capable of detecting an incorrectly configured AP, intruder, normal user and managed AP and some other rogue AP. Some of these products used scanning through sniffers like AirMagnet [8], NetStumbler [9], Aruba Networks [10] and some of them automate the process using sensors like Motorola AirDefence solutions [11], Airwave [12] Proxim [13] and Cisco Wireless LAN Solution Engine (WLSE) [14]. Automatic scanning using sensors is less time consuming than manual scanning and provides a continuous vigilance to rogue APs. Researchers have recently proposed to turn existing desktop computers into wireless sniffers to further reduce deployment cost while still providing a reasonable coverage [15]. These approaches fail when a rogue AP spoofs characteristics such as the MAC address and Service Set Identifier (SSID) of a legitimate AP. AirDefence [11] uses a combination of radio frequency sensors and an intrusion detection/protection server appliance to capture, process and correlate network events. However, these products are very costly and if the specialized monitoring sensors are not used, it is difficult to guarantee a complete coverage of the network to ensure effective rogue AP detection.

On the other hand, the research community has just recently started to direct attention toward rogue AP detection, which is discussed in next two subsections.

2.5.2 Wireless Monitoring Research

Adya et.al. [16] proposed an architecture for detecting and diagnosing faults in IEEE 802.11 for detecting rogue APs. In this scheme multiple wireless clients are instrumented to collect information about nearby APs and send the information to a centralized server for rogue APs detection. Each client is required to install special diagnostic software, and rogue APs are assumed to transmit beacon messages and respond to probe requests. However, this approach is not resilient to spoofing as it is based on assumption that rogue APs will function properly. Bahl et.al. [15] have proposed distributed monitoring framework using desktop to have a low cost management infrastructure, named as DAIR. The main idea is to enable dense RF monitoring using USB wireless adapters attached to desktop machines for more comprehensive traffic capturing ability. This study improves upon [16] by reducing false positives/negatives. However, it relies on proper operation of a large number of wireless devices, which can be difficult to manage and still relies on AP functionality that can be easily turned off. Additionally both schemes [16] and [15] assume that some specific characteristics of IEEE 802.11 standards cannot be violated by the adversaries. Multiple network sniffers are used by Chirumamilla et.al. [17] for detecting RAPs and eavesdroppers. Each sniffer has three network cards, and the IDS capabilities are thwarted by MAC address spoofing. Yeo et.al. [7] improve the performance of wireless monitoring by merging packet captures from multiple network sniffers and carefully selecting sniffer placement. The techniques are exploited to characterize MAC layer traffic and perform retrospective diagnosis.

Joshua et.al. [18] detect unauthorized access by identifying MAC address spoofing by monitoring the sequence number field in the IEEE 802.11 frame header (explained in section 2.1) as a parameter to characterize normal behavior of a wireless LAN. The sequence number field is a sequential counter that is incremented by one for each non fragmented frame. An attack is identified by a large gap in sequence numbers for an active MAC address. This way it detects MAC spoofing. However, this method can be evaded if the attackers is able to set the sequence number field to an arbitrary value. Frame loss and out of order frames also

cause false positives in sequence number checking. It also fails if the attacker is not transmitting simultaneously with legitimate device. Li et.al. [19] introduced a security layer that is separate from conventional network authentication methods. They developed forge-resistant relationship based on packet traffic by using packet sequence numbers, traffic interarrival, one-way chain of checks to detect spoofing attacks. Chen et.al. [20] have proposed K-means cluster analysis of RSS and then showed how their detector can be integrated into real-time indoor system.

In recent work, fingerprinting techniques have emerged as possible alternatives to counter the identity problem in wireless LANs. Essentially, fingerprinting is a process by which a machine, driver or the software the machine is running, can be uniquely identified due to its externally observable characteristics. Radio frequency fingerprinting captures the unique characteristics of the RF energy of a transceiver. When a radio transmitter is placed in transmit mode, a transient is generated by the frequency synthesizer whose function is to generate the carrier frequency used for transmission. It has been determined that the turn-on transients generated are distinct enough that positive identification of the transmitter is possible. Hall et.al. [21] have extended this approach to control access amongst Bluetooth wireless devices with future plans of including 802.11 transceivers. To implement this technology in a wireless LAN, special equipment for processing RF signals would be required at each AP. The cost of new equipment can become prohibitive especially for large networks with many APs. Corbett et.al. [22] have used spectral analysis to distinguish between network cards manufactured by different vendors using rate-switching algorithm that is vaguely specified in the 802.11 specification. An empirical analysis was conducted at a local hotspot to characterize rate switching. The results of this work were a unique PSD for each card that can be used as a spectral fingerprint. Though this approach proved effective, it should be noted that it assumes some level of RF interference. Faria et.al. [23] and Ureten et.al. [24] proposed to identify attackers by means of RSS measurements only. An RSS-only method simple uses the RSS measured at multiple receivers as a feature vectors. Other fingerprinting techniques that require just standard computing equipment, have been proposed by Franklin

et.al. [25]. Cache [26] has used duration field as fingerprint.

The use of clock skews of devices on a network for the purpose of fingerprinting those devices was first studied by Kohno et al. [27] and they focused on the measurement of skews in wide-area networks by observing timestamps in TCP and ICMP packets. On the wireless side, Jana et.al. [28] studied similar approach for MAC layer of 802.11 networks based of beacon timestamps.

2.5.3 Wired Monitoring Research

Differences in inter-packet spacing between IP traffic flows on wired and wireless networks used in [29], [30] and [31]. Beyah et.al. [30] demonstrated from experiments in a local testbed that wired and wireless connections can be separated based on RTT by visually inspecting the timing in the packet traces of traffic generated by the clients. The setting of their experiments are very restrictive and assumes that APs will be connected within one hop to a switch monitoring the traffic. Furthermore, the visual inspection method cannot be carried out automatically. Mano et.al. [29] use a distinctive approach based on RTT for segregating network types, complete with traffic conditioning to eliminate noise. However, it demarcates wired and wireless traffic with the help of mean and standard deviation of the RTT dataset which is not advisable as these parameters differ with the varying types, speeds and congestion levels of networks. Their approach is claimed to be non-intrusive. However since it involves conditioning of traffic, it is still, as minimum, pseudo-active. Baiamonte et.al. [32] create a spectral profile of WLANs based on the entropy of inter-arrival times in offline manner. They assume link quality and unpredictability of the wireless medium as the cause for greater wireless uncertainty. However, the scheme does not differentiate between wireless traffic from authorized and unauthorized APs. Yin et.al. [33] detect protected layer 3 rogue APs which are protected by security measures by using combination of a verifier and wireless sniffers on the internal wired network. In this approach, verifier is employed to send test traffic towards wireless edge. Once wireless sniffers capture an AP relaying the test

packets, the AP is flagged as rogue. In addition, binary hypothesis techniques are adopted to improve the robustness of detection.

Two passive online rogue AP detection algorithm were proposed in [2] by Wei et.al. The core of these algorithms are the sequential hypothesis testing applied to packet-header data which are passively collected at a monitoring point (as building gateway). Their idea is that two consecutively sent packets, such as TCP ACKs, will likely have longer time intervals between them after they traverse through a 802.11 link than through Ethernet. Once TCP ACK-pairs are observed, prompt decisions are made with little computational and storage overhead. In near future, as 802.11n APs become more popular. Its bandwidth is comparable to 100 Mbps Ethernet, making the time interval information less useful.

Ma et.al. [34] propose hybrid frameworks consolidating the above mentioned wired and wireless-side detection models for commodity Wi-Fi networks. The rogue AP protection system comprises of packet collector, rogue AP preemption and detection components. Based on availability of hardware and software resources on an AP, these components can be installed on a single AP or on separate devices connected to AP in a plugin fashion. However, real-worlds empirical results have not been provided to justify the claims of rogue AP.

2.6 Our Approach

As discussed in Section 2.2, the WLANs suffer from a number of security vulnerabilities; out of which the ability to spoof a WLAN node's MAC address is the most serious one. MAC spoofing allows an adversary to assume the MAC address of another WLAN node and launch attacks on the WLAN using the identity of the legitimate node. Without this vulnerability, an adversary will not be able to inject forged frames (management, control and EAP) into the WLAN and all attacks based on injection of such frames would be impossible. Some of these attacks are *man-in-the-middle, session hijacking, rogue AP, EAP based DoS attacks, management and control frame based DoS attacks.* To further exacerbate the problem, almost

all WLAN hardware provides a mechanism to change its MAC address; hence trivializing changing identities. A number of different techniques have been suggested to detect MAC spoofing activity in a WLAN as *sequence number monitoring, RSS, fingerprinting* as discussed in Section 2.5. Sequence number techniques is not recommended as discussed. Using MAC layer and PHY fingerprinting is a promising new area for detecting MAC spoofing. However it is not clear how reliable each fingerprint can be matched to a MAC address. It will be easy to evade as fingerprint if adversary is an insider and uses same hardware and software as that of legitimate. As similar to sequence number tracking, RSS fingerprints are temporal in nature and not fundamentally linked to APs. Thus if legitimate AP changes its location while it is switched off, its legitimacy will be difficult to truly establish thereafter.

Our approach presents a statistical based hybrid framework for rogue AP detection, which combines distributed monitoring devices to keep an eye on 802.11 link layer suspicious activities and centralized detection module at gateway router to achieve higher accuracy in detection of rogue devices as shown in Figure 2.4. This detection approach is scalable and non-intrusive and does not require any specialized hardware. It is designed to utilize the existing wireless LAN infrastructure and independent of 802.11a/b/g/n. It works on passive monitoring of wired and wireless traffic and hence easy to manage and maintain. Additionally, this approach only requires few number of packets for detection in a heterogeneous network comprised of wireless and wired subnets.

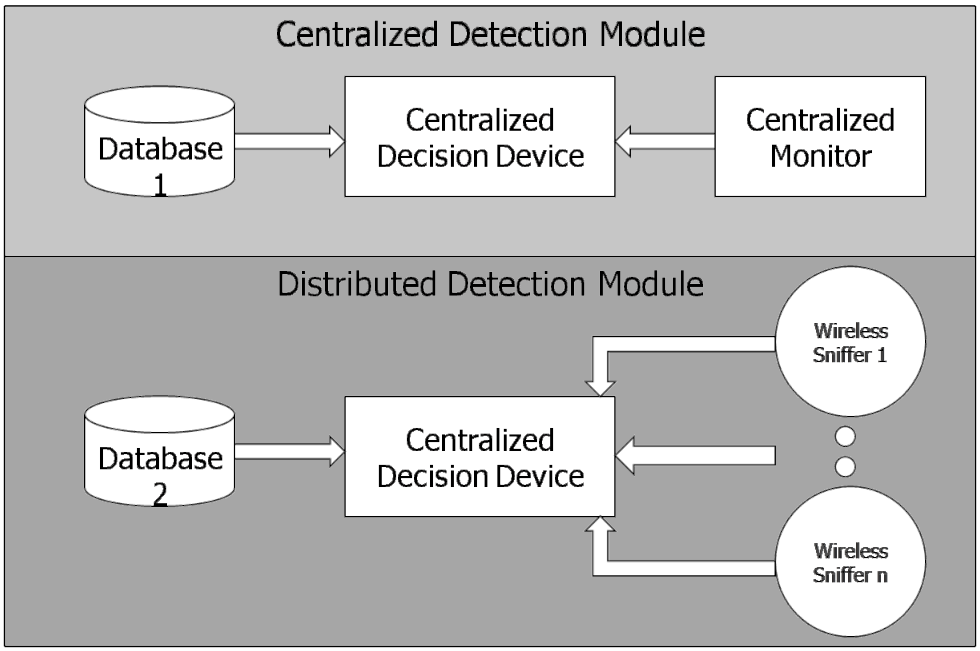


Figure 2.4: Hybrid Framework

Centralized detection is done at gateway router by differentiating wired and wireless TCP traffic by applying Weighted Sequential Hypothesis testing on inter-arrival time of TCP ACK-pairs. Decentralized module takes care of detection of MAC Spoofing and totally relies on 802.11 Beacon frames. Detection is done through analysis of clock skew and RSS, which are fingerprints of a device together with statistical methods and naïve Bayes classifier to detect presence of Rogue AP.

Chapter 3

Statistical Analysis of Wired Traffic

The presence of a wireless network infrastructure within a network raises many security issues–e.g., insertion of rogue or unauthorized APs, eavesdropping of wireless communications, attacking APs, etc. The first step in thwarting most of these attacks is identifying "malicious" traffic. Most of the existing work in this area adopts the approach of distributed monitoring of RF airwaves. However, this approach is effective at spotting rogues, but those not within the surveillance coverage range may run away undetected [34]. Therefore, the ideal method of detecting such rogue APs is to monitor traffic at a centralized aggregation point. Another benefit of a central point detection is that it alleviates the need to walk through the facilities in the case of incomplete coverage. It can be regarded as a compensation to the wireless monitoring module. This monitoring point is located at the edge of a local network (e.g., a gateway router) and captures all traffic coming into and getting out of the local network. One of the technological challenges in using this approach is detecting wireless traffic–i.e., distinguishing wireless traffic from wired traffic. Recall that a typical aggregation point observes a mixture of wired and wireless traffic, assuming that a local network supports both Ethernet and WLAN technologies. The detection of wireless traffic at the aggregation point is a very difficult task. This task cannot be accomplished by using

IP addresses. This is because a network administrator typically does not allocate separate IP address pools for wired and wireless hosts. Even if separate pools exist, a wired host may act as a NAT box for wireless hosts or it may connect to the LAN through a wireless router.

We propose a method that exploits the fundamental properties of the 802.11 CSMA/CA MAC protocol and the half-duplex nature of wireless communication channels. There is some existing work in this area that utilizes sequential hypothesis tests to detect wireless traffic [2]. Our aim is to improve the accuracy of detection of this approach while addressing the drawbacks of existing approaches. The contributions of this chapter are:

- An algorithm for *weighted sequential hypothesis testing* is developed to distinguish the wired and wireless traffic.
- Operating System (OS) diversity (specifically Windows) makes it difficult to cleanly distinguish between wired and wireless hosts. So we have incorporated OS diversity in differentiation which was not addressed in [2].
- We have incorporated latency also. Latency can be viewed as the result of either WAN-side or LAN-side effects.

This chapter begins with Section 3.1 which explain the use of inter-ACK time to differentiate wired and wireless traffic. Then Section 3.2 discusses that how a operating system can affect the hypothesis in this detection. Section 3.3 implements the proposed algorithm and how weight is given to each instance of inter-ACK time. The last Section 3.4 of this chapter gives high level overview of the detector module and how it helps in detecting rogue AP after differentiating the wired and wireless traffic.

3.1 Analysis of TCP ACK-Pairs

End hosts within any network either use Ethernet or IEEE 802.11 WLAN to access Internet. In our analysis, we are using *inter-ACK time* as statistics which can effectively differentiate between Ethernet and WLAN hosts, which inherently utilizes the intrinsic characteristics of Ethernet and WLAN. A monitoring point is located at the building gateway level or at access router of the local network (LAN), capturing traffic coming in and going out of the network, as shown in Figure 3.1, where monitoring point is co-located with access router. For each TCP flow, we identify pairs of TCP data packets destined to end host (receiver) within the local network and arriving at the monitoring point. A pair of ACKs in response to these data packets, which is known as *ACK-pairs*, are generated by the receiver and returned to the sender. Generally, the inter-arrival times of ACK-pairs at the monitoring point differ significantly if the data packets and ACK-pairs traverse a wireless hop as compared to a wired Ethernet link as shown in [2] and is demonstrated in Figure 3.1. As we see, there is less difference between inter-ACK times in Ethernet (as shown in green color) as compared to inter-ACK times which have traversed through WLAN link (shown in blue color). This difference is due to the intrinsic characteristics of Ethernet and WLAN. Our scheme which is based on timing difference of inter-ACK pairs, exploits this difference to segregate WLAN and Ethernet TCP flows.

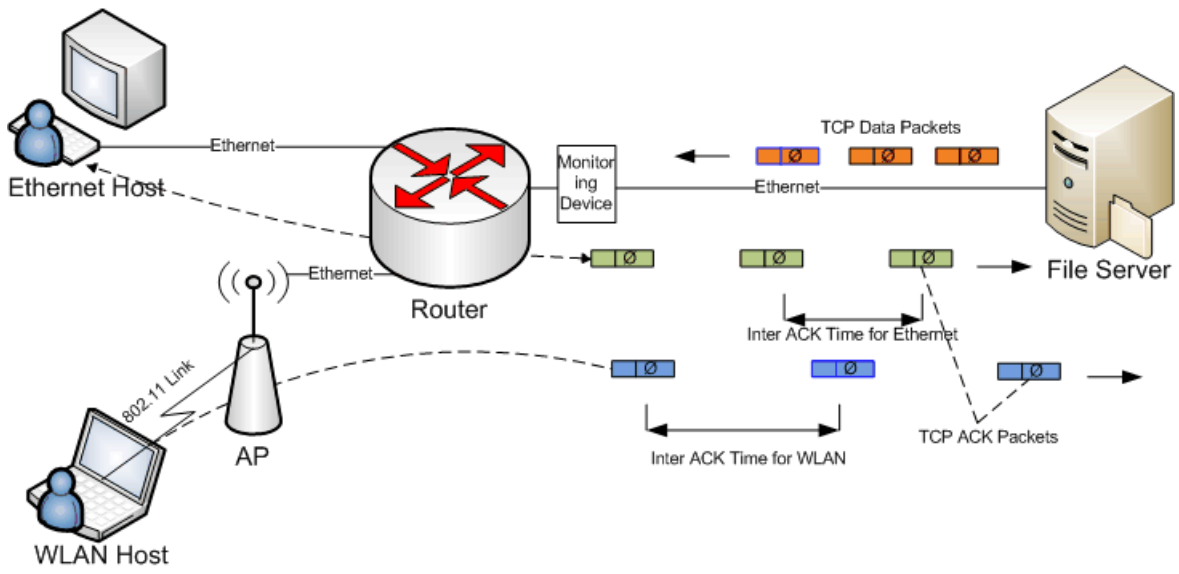


Figure 3.1: Inter ACK Time difference for Ethernet(Green) and WLAN(Blue)

As shown in Figure 3.1, an outside server sends data to a receiver residing in the LAN. The access link of the end-host (receiver) is either Ethernet or WLAN. In this setting, a router resides between the sender and the receiver, and is connected to sender over Ethernet link. Similarly, monitoring device resides between router and sender (server) over the Ethernet link. Again, user1 using desktop (receiver 1) is connected to router over the Ethernet link and termed as *Ethernet end-host*. The user2 using laptop (receiver 2) is termed as *WLAN end-host*, an AP resides between the router and receiver 2. The AP and the router are connected over the Ethernet link and the receiver 2 is connected to the AP either by using IEEE 802.11b or 802.11g (Note here we did not do the same analysis for the IEEE 802.11n). Let T denote the inter-arrival time of two data packets (data packets in Figure 3.1 are shown in orange color) that arrive at the monitoring point [2]. The receiver returns a pair of ACKs corresponding to these two data packets (ACK-pair) to the sender. Let T_A denote the inter-arrival time of this ACK-pair at the monitoring point, referred as an *inter-ACK time*.

We deduce that, when the receiver uses a WLAN, T_A is larger as compared to when receiver

uses Ethernet. Due to, the *random backoff* mechanism of IEEE 802.11 (See Section 2.1) in which host must wait for a random backoff interval to transmit and the *half duplex nature of wireless channels* (i.e., data packets and ACKs contend for media access at a wireless host) may lead to larger inter-ACK times in WLAN that those in Ethernet [2]. Although an Ethernet connection uses shared media, the randomness caused by the shared media in Ethernet is negligible as compared to the one in a wireless network because of its high bandwidth and ability to detect collisions. Authors of [2] did analysis of Ethernet and 802.11b/g traffic and proved the following:

- **For Ethernet:** Probability of T_A exceeding 600μs is $P(T_A > 600\mu s) < 0.18$.
- **For IEEE 802.11b:** Probability of T_A exceeding 600μs is $P(T_A > 600\mu s) > 0.96$.
- **For IEEE 802.11b:** Probability of T_A exceeding 600μs is $P(T_A > 600\mu s) > 0.45$.

The above analysis demonstrates that, even when WLAN is under idealized conditions while Ethernet LAN is fully utilized, using TCP ACK-pairs can effectively differentiate Ethernet and WLAN connections [2]. This analysis is based on the fundamental properties of IEEE 802.11 CSMA/CA MAC protocol (See Section 2.1) and the half-duplex nature of wireless channels. In this analysis, all retransmitted packets are excluded for better reflection of the characteristics of the access link.

3.2 Role of the Operating System in ACK Transmission

Authors of [29] showed that *Operating System (OS)* diversity (specially Windows) make it difficult to cleanly distinguish between wired and wireless hosts. Even latency plays an important role in differentiating both traffic and can be viewed as a result of either WAN-side or LAN-side effects. WAN-side latency is the result of many factors which can

vary significantly between on-going sessions and especially between differing communication host pairing [29]. On the other hand, LAN-side latency represents a reasonably consistent path between the end-host and WAN gateway. Hence we only monitor LAN-side traffic to remove WAN side latencies. Different operating systems differ in providing ratios varying from one to one and two to one depending on inter-packet spacing (Windows, Linux) with others cumulatively sending ACKs in a time-wise manner (MAC OS) [29]. Hence, operating systems that bias heavily towards cumulative ACKs would invalidate the entire foundation of differentiating Ethernet and WLAN hosts, To that end, we incorporated *Nmap* [35], which can detect the OS of the end-host and helps to build different training sets for each type of OS. That in return, provides more accurate results for detecting Ethernet or WLAN hosts.

3.3 Weighted Sequential Hypothesis Testing

In this section, we develop an algorithm using *weighted sequential hypothesis test* to detect wireless hosts based on analysis which is given is Section 3.1. This algorithm is an improvement proposed on the work of [2] where they have used sequential hypothesis test [36], in which the size of sample is not fixed and data is evaluated as it is collected. Our algorithm uses weighted sequential hypothesis test technique and takes inter-ACK times as the input.

Let define two hypothesis H_0 and H_a, representing respectively the *null hypothesis* that a host uses Ethernet and the *alternative hypothesis* that the host uses WLAN. To apply the weighted sequential hypothesis test, we need to know the inter-ACK time distributions for Ethernet and WLAN beforehand. In general, the inter-ACK time distribution for a connection type can be acquired from a training set, which contains TCP flows known to use this connection type. We first train the algorithm with the calculation of *Cumulative probability Density Function (CDF)* for both Ethernet and WLAN scenarios. When these distributions are known, we can calculate the likelihoods that a host uses Ethernet and WLAN respectively given a sequence of observed inter-ACK times. If the likelihood of using

WLAN is much higher than that of using Ethernet, we conclude that the host uses WLAN and vice versa [2].

Now we explain the algorithm in detail. Let $\{t_i^A\}_{i=1}^N$ represent a sequence of inter-ACK time observations from a host, and $\{T_i^A\}_{i=1}^N$ represent their corresponding random variables. Let E and W represent respectively the events that a host uses Ethernet or WLAN. We assume that the inter-ACK times are independent and identically distributed then,

$$L_E = P(T_1^A = t_1^A, T_2^A = t_2^A, T_3^A = t_3^A, \cdots, T_N^A = t_N^A \mid E) = \prod_{i=1}^{N} p_i \tag{3.1}$$

L_E be the likelihood that the observation sequence is from an Ethernet host, where p_i is the probability that the ith inter-ACK time has value t_i^A given that it is from an Ethernet host and calculated from Ethernet CDF as:

$$p_i = P(T_i^A < (t_i^A + \delta_E^A) \mid E) - P(T_i^A < t_i^A \mid E) \tag{3.2}$$

where δ_E^A is the smallest granularity for the calculation of p_i. Similarly,

$$L_W = P(T_1^A = t_1^A, T_2^A = t_2^A, T_3^A = t_3^A, \cdots, T_N^A = t_N^A \mid W) = \prod_{i=1}^{N} q_i \tag{3.3}$$

L_W be the likelihood that the observation sequence is from a WLAN host, where q_i be the probability that the ith inter-ACK time has value t_i^A given that it is from an WLAN host and calculated from WLAN CDF as:

$$q_i = P(T_i^A < (t_i^A + \delta_W^A) \mid W) - P(T_i^A < t_i^A \mid W) \tag{3.4}$$

where δ_W^A is the smallest granularity for the calculation of q_i.

Both p_i and q_i are obtained from the inter-ACK time distributions of Ethernet and WLAN respectively from training dataset. The values of L_E and L_W are updated when a new ACK-

pair is observed. Let $K > 1$ be the threshold, then if after n^{th} ACK-pair, the likelihood ratio of L_E and L_W is over the threshold ($\frac{L_W}{L_E} > K$), then we reject null hypothesis H_0 and the host is classified as an WLAN host. If ($\frac{L_W}{L_E} < \frac{1}{K}$), then we accept null hypothesis H_0 and the host is classified as an Ethernet host. We have used *log-likelihood* functions for implementation instead of the likelihood functions and are defined as $l_w = \log(L_W)$ and $l_e = \log(L_E)$. N is the heuristic number which shows how many number of ACK-pair required to differentiate Ethernet and WLAN hosts. In most of cases, only 10 ACK-pair utmost required to differentiate but for few cases it went till 20. Algorithm 3.1 shows the pseudo-code for weighted sequential hypothesis test.

Algorithm 3.1 Weighted Sequential Hypothesis Test

Require: inter-ACK time from training and test data
Ensure: Classifies hosts as Ethernet or WLAN

```
Compute the CDF for both Ethernet and WLAN from training data
Compute δ_E^A from Ethernet CDF
Compute δ_W^A from WLAN CDF
Get the test data
n ← 0, l_E ← 0, l_W ← 0
while n ≠ N do
    Identify an ACK-pair
    n ← n + 1
    p_n ← P(T_n^A = t_n^A | E)
    q_n ← P(T_n^A = t_n^A | W)
    if p_n > q_n then
        e_n^A ← P(T_n^A = t_n^A + δ_E^A | E) / P(T_n^A = t_n^A | E)
        w_n^A ← 0
    else if q_n > p_n then
        w_n^A ← P(T_n^A = t_n^A + δ_W^A | W) / P(T_n^A = t_n^A | W)
        e_n^A ← 0
    else
        e_n^A ← 0, w_n^A ← 0
    end if
    l_E ← l_E + log(p_n) + log(1 + e_n^A)
    l_W ← l_W + log(q_n) + log(1 + w_n^A)
    if l_W − l_E > log(K) then
        n ← 0, l_E ← 0, l_W ← 0
        return WLAN
    else if l_W − l_E < −log(K) then
        n ← 0, l_E ← 0, l_W ← 0
        return Ethernet
    end if
end while
n ← 0, l_E ← 0, l_W ← 0
return Host Undetermined
```

For the detection of a given test data, first the probabilities of inter-ACK time for Ethernet and WLAN are calculated from CDF as p_n and q_n respectively. In a case when either of the probability is smaller gives us information that for specific case the probability for one

of the event is higher than the other probability. In such case we make a *biased* judgement and assign a weight as:

If($p_n > q_n$),

$$e_n^A = \frac{P(T_n^A = t_n^A + \delta_E^A \mid E)}{P(T_n^A = t_n^A \mid E)} \tag{3.5}$$

If($q_n > p_n$),

$$w_n^A = \frac{P(T_n^A = t_n^A + \delta_W^A \mid W)}{P(T_n^A = t_n^A \mid W)} \tag{3.6}$$

If($p_n = q_n$),

$$e_n^A = w_n^A = 0 \tag{3.7}$$

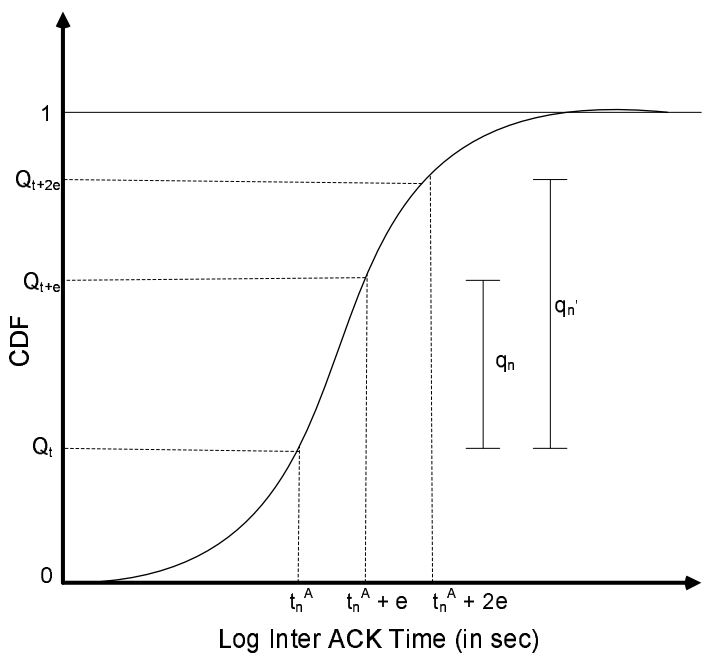


Figure 3.2: Adaptive Weight Assignment for Sequential Hypothesis Testing

After the calculation of respective adaptive weight of both Ethernet and WLAN, we can modify the probabilities for Ethernet and WLAN by:

$$
\begin{aligned}
p_n &= P(T_n^A = t_n^A \mid E) \\
&= P(T_n^A < t_n^A + \delta_E^A \mid E) - P(T_n^A < t_n^A \mid E) \\
p_n^{'} &= P(T_n^A < t_n^A + 2\delta_E^A \mid E) - P(T_n^A < t_n^A \mid E) \\
p_n^{'} - p_n &= P(T_n^A < t_n^A + 2\delta_E^A \mid E) - P(T_n^A < t_n^A + \delta_E^A \mid E) \\
p_n^{'} &= p_n + P(T_n^A < t_n^A + 2\delta_E^A \mid E) - P(T_n^A < t_n^A + \delta_E^A \mid E) \\
p_n^{'} &= p_n + P(T_n^A = t_n^A + \delta_E^A \mid E) \\
\text{assume } P(T_n^A = t_n^A + \delta_E^A \mid E) &= e_n^A p_n \text{ where } e_n^A \geq 0 \\
p_n^{'} &= (1 + e_n^A) p_n \\
\text{Similarly, } q_n^{'} &= (1 + w_n^A) q_n \\
\text{where } P(T_n^A = t_n^A + \delta_W^A \mid W) &= w_n^A q_n \text{ and } w_n^A \geq 0
\end{aligned}
$$

Figure 3.2 shows graphical interpretation of adaptive weight assignment and shows how q_n is modified to $q_n^{'}$ by incrementing time from $t_n^A + e$ to $t_n^A + 2e$. As CDF is always monotonically increasing function, we will always get the improvement as $q_n^{'} \geq q_n$. After the respective probabilities are calculated for all the test values, we evaluate l_E and l_W as described in Algorithm 3.1 which in turn helps in determining whether the given test data belongs to WLAN host or Ethernet host.

3.4 Detector Module and Detection Methodology

In this section, we present the design of the detector module and how it helps to detect rogue AP. This module consists of below major components as shown in Figure 3.3.

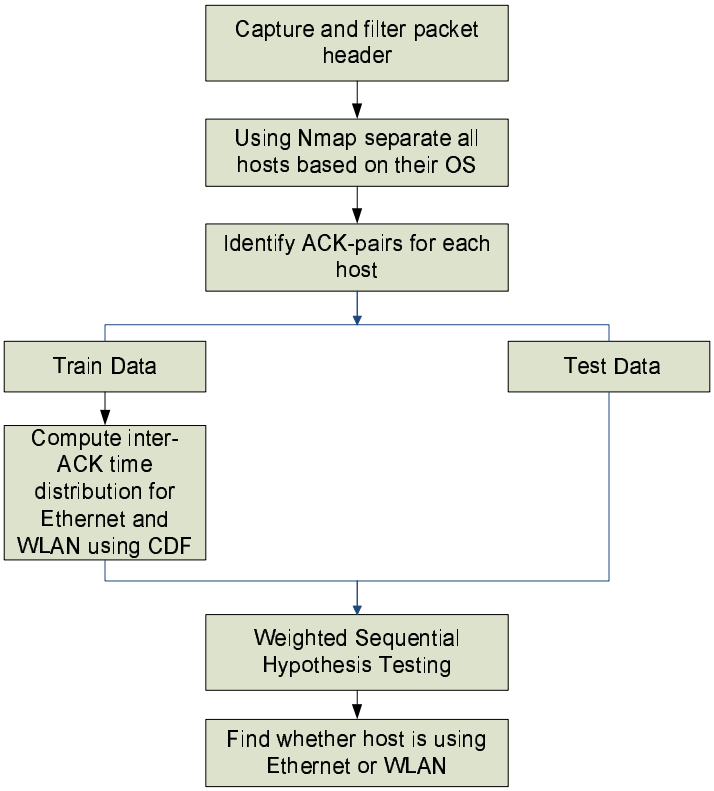


Figure 3.3: High Level Overview of Detector Module

The data capturing component is needed for realtime WAN traffic monitoring so that rogue device can be quickly identified using algorithm as described in Section 3.3. This component collects incoming and outgoing packets and extract header information from packets. Then with the help of Nmap [35], it tries to find out the OS for each hosts and makes separate databases based on OS of the host. Note that these databases are not created based on Ethernet or 802.11, it purely uses the information of the OS, which end-host is using. It might be possible that a single host is using many TCP flows instantaneously. Detector module maintains a record for each TCP flow. Then for each flow per host ACK-pairs are identified. Then some sets of ACK-pairs are used for training purpose and some are used for testing purpose. The training set for Ethernet or WLAN is constructed by extracting TCP flows destined to hosts in Ethernet or WLAN from a trace collected at the monitoring point. From training sets, we compute the CDF for both Ethernet and WLAN from the sequences

of ACK-pairs. Then we calculate inter-ACK time distribution based on CDF for Ethernet and WLAN traffic. Then from test data set to validate that our algorithm can detect WLAN hosts while does not misclassify Ethernet hosts, ACK-pairs are fed into weighted sequential hypothesis testing module to determining whether the host uses Ethernet or WLAN.

Once a WLAN host is detected, its IP address can be looked from an authorization list [2] for rogue AP detection. If IP mismatch is found, then based on host's IP, access router of this host is determined. Then from ARP table of access router, MAC address of the WLAN host can be determined. From the MAC address of the host, its corresponding switch port is determined using SNMP queries. Lastly, sequential queries to downstream switches are used to locate switch port from which rogue AP is connected [2]. Evil twins can be detected by hop analysis, as legitimate AP will be used only one hop for 802.11 traffic but evil twin will be using two hops and can be detected through traffic analysis. Other common activity which is done by rogue AP is the port scanning, in which unauthorized WLAN host performs a port scanning operation to find the end hosts with vulnerabilities. This can be detected if the frequency of straight-access and crossing-access exceeds a nominal threshold [31].

Chapter 4

Statistical Analysis of Wireless Traffic

As MAC address can be easily changed through software, simple yet effective identity-based attacks can be done with off-the-self equipment against multiple link-layer services. As discussed in Chapter 2, since control and management frames can be spoofed and forged even with WPA2 and 802.11i, WLANs remain susceptible to identity attacks and DDoS attacks. Without this vulnerability, an adversary will not able to inject forged management, control and EAP frames into the WLAN and all attacks based on injection of such frames would be impossible. Even exploiting the unprotected MAC frame *duration field* to cause a virtual jamming is also only possible in combination with MAC spoofing. Hence, MAC spoofing is responsible for a majority of the attacks on WLANs. Therefore it is necessary to develop techniques that will allow us to detect MAC spoofing by impersonating MAC frames.

Detection of such impersonation could lead us to the detection of a great variety of attacks. In this chapter, we propose a new technique to detect the MAC spoofing. Section 4.1 explains how fingerprinting techniques can be used to differentiate between legitimate APs and rogue APs. It also gives the outline to our fingerprinting technique for rogue AP detection. Section 4.2 presents definition of clock skew and methodology for measuring clock skews of APs. The following Section 4.3 describes how RSS can be used for fingerprinting and how is it measured.

Lastly, in Section 4.4, we show how to use the naïve Bayes classifier to classify the different APs using clock skew and RSS.

4.1 Fingerprinting of IEEE 802.11 for Mac Spoofing

In recent work, fingerprinting techniques have emerged as possible alternatives to counter the problem of identity in WLANs. Essentially, fingerprinting is a process by which a machine, driver or the software run on the machine can be uniquely identified due to its externally observable characteristics. Such fingerprinting has innumerable applications. Fingerprints enable us to identify IEEE 802.11 implementations. Fingerprints can also be used in a defensive way. For instance it is useful from the point of view of network forensics for identification purposes by maintaining a database of authorized devices approved for use on their WLAN. After knowing which 802.11 implementations are vulnerable, an administrator can monitor their environment for wireless activity, observe 802.11 fingerprints, and find out the presence of rogue APs.

Developing 802.11 fingerprints is largely an exploratory exercise in determining how an 802.11 implementation behaves uniquely. The strength of a fingerprint determines whether the components of an implementation can be identified individually. The fingerprints described in this research perform reliable identification of device based on their hardware clocks and their locations. Before getting involved in complex cryptographic exchange protocols (like WPA) with an untrusted entity, fingerprinting the AP can be a mean of providing a first point of trust for clients. Once the fingerprints of the AP are verified, the clients can proceed with cryptographic protocols to reaffirm their trust.

Our main contribution to this research is to propose a new wireless fingerprinting technique that differentiates between unique devices through timing analysis of 802.11 beacon frames and Received Signal Strength (RSS). Using MAC layer and PHY fingerprinting is a new and promising area for detecting MAC spoofing. So in our case, a unique device is defined as a

unique combination of *(clock skew (at MAC layer) and RSS (at PHY layer))*. This technique is passive, accurate, robust and noninvasive and can be used as a complement to the existing authentication methods. Our method is based on naïve Bayes classification which is an example of statistical analysis of the timing information in beacon frames. A timing-based approach has a number of advantages over a content-based approach as described in Section 2.5. Primarily among these is the fact that coarse-grained timing information is preserved despite the encryption of frame content using WEP and WPA.

4.2 Measurement of Clock Skew

Minute deviations between the clock oscillators of different machines could result in clock skews [37]. Clock offset is the difference in time, and *clock skew* is the difference in clock speeds of sender (in our case AP) and receiver (fingerprinter). Clock skews have also been found to be relatively stable and constant [38]. We define the following terms related to clock from [38]. Let C_a and C_b be two clocks then,

- **Clock Frequency:** It is the rate at which the clock progresses and given at any time $t \geq 0$ of C_a as $C_a^{'}(t)$
- **Clock Offset:** The offset of the clock C_a relative to C_b at time $t \geq 0$ is $C_a(t) - C_b(t)$.
- **Clock Skew:** It is the difference in the frequencies of two clocks. Clock skew of C_a relative to C_b at time $t \geq 0$ is given as $C_a^{'}(t) - C_b^{'}(t)$

The use of clock skews of devices on a network for the purpose of fingerprinting was first studied by Kohno et.al [27]. They showed that it is possible to remotely measure the microscopic skews of devices, and their fingerprinting methods could identify individual devices despite errors inherent in remote measurements. The study of authors focused on the measurement of skews in wide-area networks by observing timestamps in TCP and ICMP packets. On the wireless side, Jana et.al [28] and Arackaparambil et.al [39] studied the same approach of [27]

at MAC layer of 802.11 networks. Unlike the scheme in [27], they used TSF timestamps in the beacon frames sent by AP, to determine clock skew. They observed that, due to the essentially zero latency and the availability of a high frequency stream of high precision beacon timestamps, the process of measuring clock skews would become more accurate and effective in these networks. We propose a fingerprinting technique which uses the clock skew as one feature to detect rogue AP. 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 the time.

In 802.11 network, synchronization is achieved through the beacon frames transmitted by the AP at periodic intervals. The most common setting for the beacon interval is 10 milliseconds, so that beacons are commonly transmitted at the rate of 10 beacons/second independent of any application. The beacon frames contain the TSF timer timestamp (see Figure 4.1) of the AP at an instant when the data symbol of the first bit of the timestamp is transmitted to the wireless medium, adjusting for hardware transmission delays. The timer is of microsecond resolution and is maintained as a 64-bit counter. This implies that the beacon timestamps provide a high-precision mechanism to measure the skew in the TSF timer of APs.

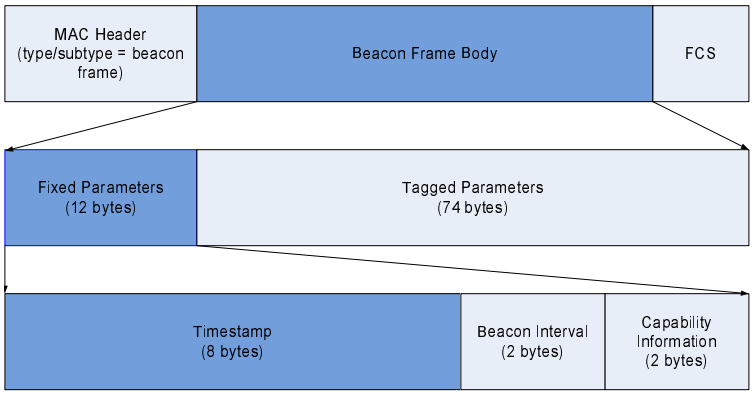


Figure 4.1: Beacon Frame Format

We now define the notion of clock skew as given by [37], [38], and the later used by [27], [28] and [39]. This is explained in Figure 4.2. To measure the clock skew of an AP, we passively

monitor the wireless interface of the fingerprinter (or monitoring device) for beacon frames from the AP. In the most of cases, the beacon frames are sent at the fixed data rate and the size of the beacon frames remain fixed [1] as shown in Figure 4.1 and hence transmission delay between AP and fingerprinter is constant for all beacon frames. We define following terms for the calculation of the clock skew. It can be explained by Figure 4.2:

- $\mathbf{C_s}$: Access Point's Clock
- $\mathbf{C_r}$: Fingerprinter's Clock
- $\mathbf{n}$: number of measured beacon frames
- $\mathbf{T_i}$: Generation time of the ith beacon frame according to C_s, it is also the timestamp conveyed by the ith frame
- $\mathbf{t_i}$: arrival time of the ith beacon frame received by fingerprinter according to C_r
- $\mathbf{x_i}$: time difference between the first received beacon frame t_1 and t_i i.e., the ith beacon frame according to the C_r
- $\mathbf{w_i}$: time difference between the first generated beacon frame T_1 and T_i i.e., the ith beacon frame according to the C_s
- $\mathbf{y_i}$: clock offset of the ith frame measurement according to the C_r

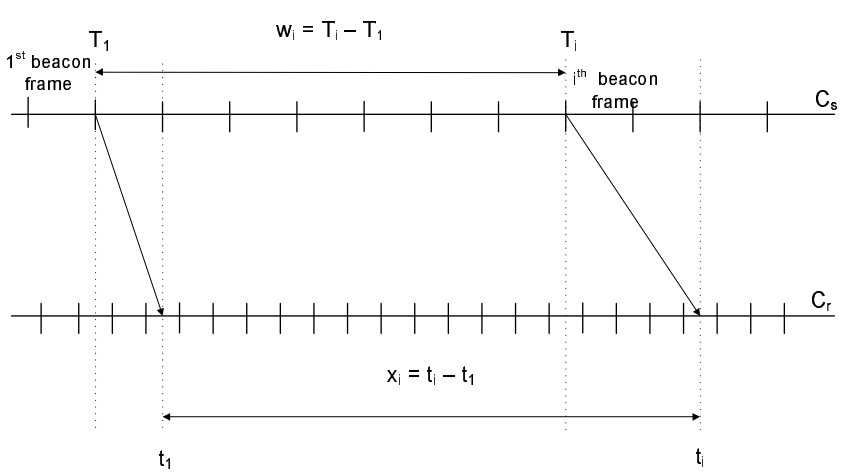


Figure 4.2: Timing Chart Showing Variable Delay

So as shown in Figure 4.2, if we have n measurements of $(t_i, T_i), 1 \leq i \leq n$, then x_i and y_i are given as (assuming fixed data rate and fixed size):

$$x_i = t_i - t_1 \tag{4.1}$$
$$w_i = T_i - T_1 \tag{4.2}$$
$$y_i = w_i - x_i \tag{4.3}$$

So, if there is no relative clock skew then we will have $w_i = x_i$ for all the measurements but in reality that is not the case. So from the above Equation 4.1, we get a set of n points (x_i, y_i) representing the clock offsets. So, if the clock skew of a particular device remains constant and if we plot (x_i, y_i), we will get approximately linear patterns. The clock skew can be measured as the slope of this linear pattern. In Figure 4.3, we have plotted clock offsets of two different APs for 700 beacon frames. After observing their linear pattern, we see that both the devices have different clock skews.

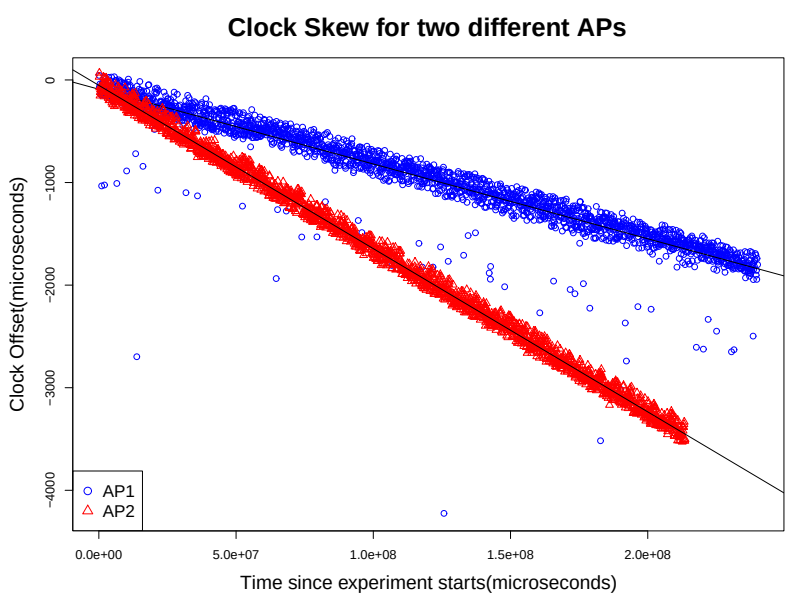


Figure 4.3: TSF clock skews for two different APs

Similar to [27], [28] and [39], we have examined two different methods for estimating the clock skew of an AP. First approach uses *Linear Programming Method (LPM)*, which finds a line that upperbounds all the time offsets calculated from the timestamps in the AP beacons and the time of arrival of those beacons at fingerprinting node. Let $y = \alpha \cdot x + \beta$ represents the line equation. The slope of this line (α) is clock skew of that AP. Second approach employs *Least Square Fitting (LSF)*, which finds a line that is at the least square distance from all the time offsets. The slope of the line represents estimated clock skew (α) of the given AP.

4.2.1 Linear Programming Method (LPM)

This methods produces a line that upper-bounds the set of clock offset points, while minimizing the sum of the distances of the points from the line $y = \alpha \cdot x + \beta$, where α is the slope of the line and β is the y-axis intercept. It outputs the slope of the line, α, as the clock skew estimate. So for the clock skew estimation, α, we have the optimization problem with constraints $\forall\ i = 1, \cdots, n$,

$$\alpha \cdot x_i + \beta \geq y_i, \tag{4.4}$$

and with the following objective function,

$$\text{minimize: } \frac{1}{n} \sum_{i=1}^{n} (\alpha \cdot x_i + \beta - y_i) \tag{4.5}$$

This method chooses an upper bound that captures the effects of outliers due to network delays. The clock skew estimate remains stable even if there were significant number of outliers. This problem can be solved using linear programming methods for two variables.

4.2.2 Least Square Fitting (LSF)

However, it was shown in [28] that number of outliers are very less in WLANs as compare to WAN because of no significant delay involved in the communication path. Additionally, TSF clocks are high precision clocks and hence linear LSF will also serve the purpose. LSF is a simple statistical linear regression method that fits the line $y = \alpha \cdot x + \beta$, where α is the slope of the line and β is y-axis intercept, to the set of clock offset points (x_i, y_i) by minimizing the least square error as,

$$\sum_{i=1}^{n} (y_i - (\alpha \cdot x_i + \beta))^2 \tag{4.6}$$

LSF was successfully applied in measuring clock skews from beacon timestamps [28]. Authors have also pointed out that it is advantageous to use LSF as compared to LPM in case when an adversary tries to avoid detection by interspersing frames from a rogue AP with the frames from the legitimate AP.

4.3 Measurement of Received Signal Strength (RSS)

Received Signal Strength (RSS) is the signal strength of a received frame measured at the fingerprinter. Fingerprinter's antenna power output, its sensitivity and path loss all contribute to signal strength. In IEEE 802.11 networks the *Received Signal Strength Indicator (RSSI)* value is used while performing medium access control clear channel assessments and in roaming operations. RSS has been widely used in indoor geographical location and positioning systems in WLANs. Indirect methods to detect spoofed MAC addresses were introduced in [23] by fingerprinting the physical location of wireless clients. Their scheme fingerprints the location of each client by measuring the RSS of client-transmitted packets with respect to n APs within its range. Despite being quite erratic, RSS values generally follow a fairly tight and predictable distribution. RSS can be used as physical layer fingerprint as legiti-

mate APs rarely alter their positions and it is possible to identify them through distinctive RSS signatures. It is hard for an attacker to modify the signal strength of his or her wireless devices during the network transmission. For this reason, the RSS at the physical layer is a good signature or fingerprint for both STA and AP. There are four units of measurement to represent the RF signal strength: mW (milliwatts), dB (decibels), percentage measurement and relative RSSI.

In the case of a spoof, the unpredictability of the propagation of RF energy is a good thing. It forces each AP to have a unique RF signature from the perspective of a fingerprinter; hence is secure from eavesdropping and is very difficult to predict. By periodically monitoring the RSS values for a particular AP from a passive fingerprinter, a dynamic RSS profile can be developed for that node. Any abrupt or unusual changes in the RSS profile for a node are indications of spoofing attack targeting that AP. This RSS profile is dynamic as it is constantly updated with the latest RSS values for the AP, as observed by the fingerprinter. Even if the rogue AP and legitimate AP were located next to each other and transmitting the same power, the paths that those emissions would take through the air would be slightly different. In addition to the uniqueness of RSSI at the fingerprinter, the rogue AP has no idea what it looks like from the fingerprinter's perspective.

By using path loss models RSS is estimated as a function of distance [40] which also include shadowning. Both theoretical and measurement-based propagation models indicate that average RSS decreases logarithmically with distance, whether in outdoor or indoor radio channels. The average large-scale path loss for an arbitrary transmitter-receiver separation is expressed as a function of distance by using a path loss exponent, n,

$$\overline{PL}(dB) = \overline{PL} + 10n\log(\frac{d}{d_0}) \tag{4.7}$$

where, n is the path loss exponent which indicates the rate at which the path loss increases with distance, d_0 is the close-in reference distance which is determined from measurements close to the transmitter, and d is the transmitter-receiver separation distance. The bars in

Equation 4.7 denote the ensemble average of all possible path loss values for a given value of d.

What makes the use of RSSIs difficult in practice is that there is always environmental variation, calibration drift and the other factors that make tracing one device's RSSIs unstable and noisy. However, for the most part RSSIs from frames originating in the same device are normally distributed and hence, statistical analysis can be done on this [41]. The main limitation of this spoof detection technique is its association of identity to physical location. If a legitimate client changes its location while it is switched off, its legitimacy will be difficult to truly establish thereafter which will not happen in case of AP. On the other hand, methods that employ the characteristics of the RF channel are limited in their resolution by the wavelength of wireless technology, which can be of the order of tens of centimeters for WLANs.

Our fingerprinting technique aims to resolve the problem of identity resolution between 802.11 APs, by distinguishing between unique combinations of clock skew and RSS. The key idea of our technique is to distinguish between unique devices by timing analysis of beacon frames and RSS. This analysis is done through naïve Bayes classifier.

4.4 Naïve Bayes Classifier

Research in the theory of machine learning has been developed for several years and there is a variety of different methods and tools, each developed to solve a particular problem in certain area of science. This research considers *naïve Bayes* method due to its simplicity and effectiveness in application traffic classification [42]. It was also used for user fingerprinting in [43]. Naïve Bayes gives a simple approach, with clear semantics, to representing, using and learning probabilistic knowledge. It is called naïve Bayes because it is based on Bayes's rule and "naïve" assumes independence – it is only valid to multiply probabilities when the events are independent [44]. In simple terms, a naïve Bayes classifier assumes that the

presence (or absence) of a particular feature of a class, is unrelated to the presence (or absence) of any other feature. Impressive results can be achieved using it. An advantage of the naïve Bayes classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix. It has often been shown that naïve Bayes rivals, and indeed outperforms, more sophisticated classifiers on many datasets. It is sufficient to classify legitimate and rogue APs.

We construct a classifier C which represents number of total APs in wireless networks. Assume there are total k such APs, then C is defined as $C = (c_1, \cdots, c_k)$. Given, a 802.11 traffic data sample $\mathbf{x} = (x_1, \cdots, x_n)$, c_j returns "Yes" if it believes the data sample came from access point AP_j and "No" otherwise. It means that for each observed instance x_i in $\mathbf{x}$, there is a known mapping $M : x \rightarrow C$ representing the membership of instance x_i to a particular class of AP. The notation $M(x_i) = c_j$ stands for "the instance x_i belongs to the class c_j".

Traffic sample $\mathbf{x}$ is a realization of $\mathbf{X} = (X_1, \cdots, X_n)$ such that each random variable X_i is described as m features $(f_1, \cdots, f_m)$ which are extracted from traffic sample. As every feature has different source, it can be considered as independent. In our case, we have only two features (clock skew and RSS; both have numeric values). So our traffic sample is represented by two features as f_1, f_2. So, $X_i = (f_1^i, f_2^i)$ is then a random vector.

Bayesian statistical conclusions about the class c_j of an unobserved traffic sample $\mathbf{y}$ are based on probability conditional on observing the traffic sample $\mathbf{y}$. This is called the *posterior probability distribution* and is denoted by $Pr[c_j|\mathbf{y}]$ and it is given by Bayes rule as [42]:

$$Pr[c_j|\mathbf{y}] = \frac{Pr[c_j] \cdot Pr[\mathbf{y}|c_j]}{\sum_{c_j} Pr[c_j] \cdot Pr[\mathbf{y}|c_j]} \tag{4.8}$$

where $Pr[c_j]$ denotes the probability of obtaining class c_j independently of the observed

data and it is known as *prior probability distribution* and estimated from some training set $\mathbf{x}$, $Pr[\mathbf{y}|c_j]$ is the *conditional probability distribution* as probability of $\mathbf{y}$ given c_j and the denominator acts as a normalizing constant. Simplified form of the above equation can be written as

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

Our fingerprinting technique is a process which is explained in flow chart in Figure 4.4. This is the flow chart for classification of APs. Flow starts with capturing 802.11 wireless traffic with the help of sniffer. In essence, the traffic capturing involves the passive collection of beacon frames emitted from APs for timing analysis of clock skew and we also extracted RSS (in dBm) values from *radio-tap header* [45] and this task can be performed using *tcpdump* [46]. After collecting the traffic, relevant features are extracted from both of data sets, called field parameters which come in every 802.11 frame. In our case they are mainly clock skew and RSS and for that we preprocess the capture to extract the RSS (in dBm) and arrival time of frame at fingerprinter and TSF from beacon frame. After collecting the traffic, it is divided into many samples. Some samples are used as training data and others are used as test data to analyze the accuracy of the classifier.

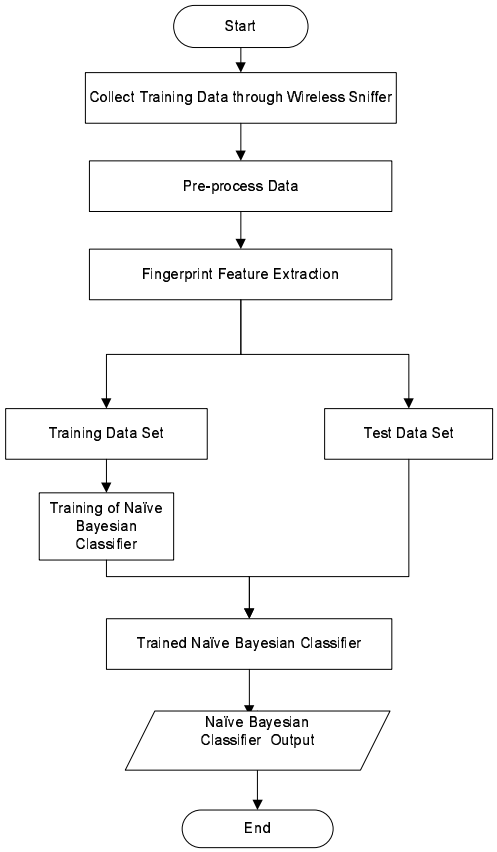


Figure 4.4: Flow Chart of Fingerprinting Technique

Then after extraction of required field, clock skew is calculated for a sample and used a single value feature for classifier which is shown as pre-processing of the data. RSS values are passed as they are collected for training of naïve Bayes classifier. Once training is done, test data is passed to classifier to test the validity of the classifier.

Chapter 5

Experimental Evaluation and Analysis

In this chapter, we present experimental evaluation and results analysis of the hybrid framework for wired and wireless traffic as described in Chapters 3 and 4 respectively. This chapter starts with Section 5.1 which discusses about the testbed used to generate dataset using various scenarios and is followed by results analysis using Algorithm 3.1 and detector module which is explained in Section 3.4. This discussion is followed by Section 5.2 for statistical analysis of wireless traffic for MAC spoofing. It also describes testbed which is used to simulate MAC spoofing attack. Under different network conditions and scenarios, diverse results were produced. Finally it discusses about the detailed analysis of various scenarios of MAC spoofing attack and presents results associated with them.

5.1 Wired Traffic Analysis and Results

For real time TCP-IP traffic analysis, we have simulated the testbed (See Figure 5.1), in which router is a centralized entity where we capture the on-going traffic for wired and wireless hosts using a monitoring device. A switch is placed between the AP and the router to form a LAN. The LAN consists of two wired and two wireless hosts. An Apache server is

connected to the router and which sends the TCP data packets to wired and wireless hosts and in turn they send TCP-ACKs (See Section 3.1). These ACKs are captured at monitoring point. Now we use algorithm 3.1, to find out transmission medium of the hosts with the help of Nmap [35].

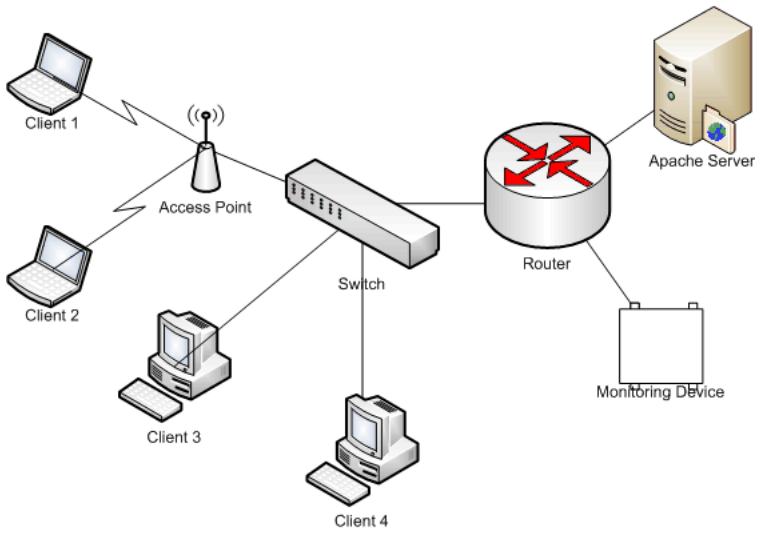


Figure 5.1: Experimental Testbed

We have used a Red Hat Linux machine with 4 Ethernet ports. One of the port (eth0) is connected to the Internet and other (eth1) is connected to Belkin wireless router. We have configured routing functionality on this machine by modifying the IP tables and constructing Dynamic Host Configuration Protocol (DHCP) server.

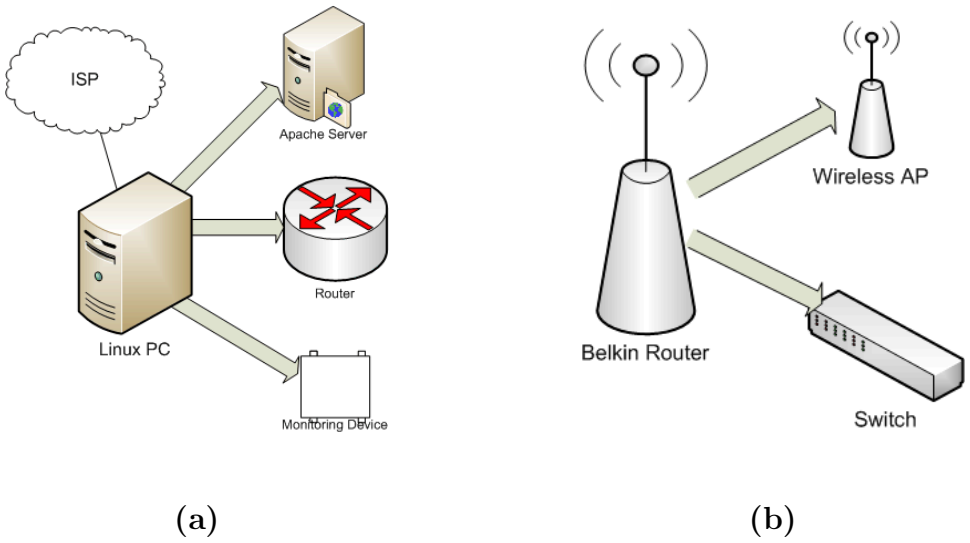


Figure 5.2: Components configured on (a) Desktop and (b) Wireless Router

On top of that, we have configured an Apache server on the same machine. Hosts create TCP sessions with Apache server and from there, they download the files. Monitoring device (we use tcpdump [46] for capturing TCP traffic) is also connected to the same machine. In similar way, we have re-configured wireless router to work as a dummy AP and switch. To do that we have switched off all routing functionalities of wireless router (See Figure 5.2). So it works only as a Layer 2 device.

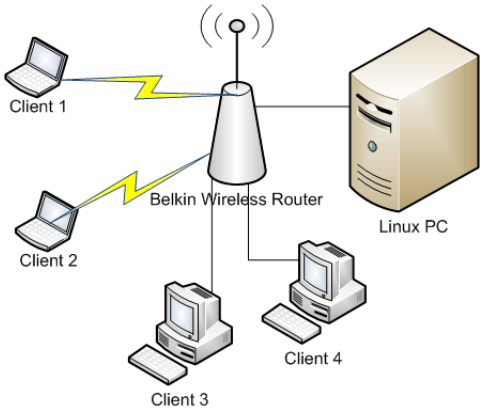


Figure 5.3: Final Testbed

So final setup is as shown in Figure 5.3, in which Linux PC has functionality of Apache server, router and monitoring device. In similar way, wireless router is acting as switch and AP. We have made a html page for Apache server, which is used by hosts to connect to server and to request for the file download. This is simulated for data flow between hosts and server. We have used around 700 Mb of iso file, which is downloaded from the server to host machines.

Analysis was done for various scenarios with the help of three machines (one desktop and 2 laptops), which simulate as many hosts. The dataset was in the form of dozens of pcap files created using tcpdump for 8-9 hours and analyzed offline. To process these data files, custom perl scripts are written, which extract the required fields and compute inter-ACK time for each TCP-ACK pair. Code of the Algorithm 3.1 is written in MATLAB [47] using $K = 10^5$ (See Section 3.3). Detector module consists of perl scripts, algorithm and Nmap. To present the results of the experiments, a detailed discussion of the various scenarios is followed by discussion of several notable cases of interest.

- In the first scenario, we have used Red Hat Linux PC as wired host and two laptops (Windows XP machines) as wireless hosts. Then, we trained the Algorithm 3.1 with wired host and one laptop wireless host. Another wireless host is used as test host. For this case, 10 sets of data were collected and analyzed. Results were up to the expectations as algorithm was able to find the wireless host with an accuracy of 100% with the help of NMap [35] because it was able to differentiate the operating system. Similarly, with all the other combinations (where trained and test data were using same OS), the algorithm is able to differentiate wired and wireless host.

- Then we have used PC (Red Hut Linux) as wired host, one laptop as wireless host (Windows XP) and again trained the algorithm 3.1 and then second laptop was used as wired host (Windows) and re-run the test again with Nmap and it failed as trained wired data was using Linux OS and test wired data was using Windows OS. This validates the assumption about OS contribution in the analysis of wired and wireless

traffic. This test has passed using Nmap.

- Generally the algorithm requires lesser than 10 ACK-pairs for classification with the accuracy of 100%. But for some cases, it needs 20 ACK-pairs to differentiate wired vs. wireless host. Twenty ACK-pairs were sufficient enough to differentiate wired and wireless host.

Hence, detector module is successfully able to differentiate between wired and wireless traffic in all cases by taking care of OS and latency. This is a very remarkable accomplishment. Figure 5.4 shows CDF of all Ethernet and WLAN training sets.

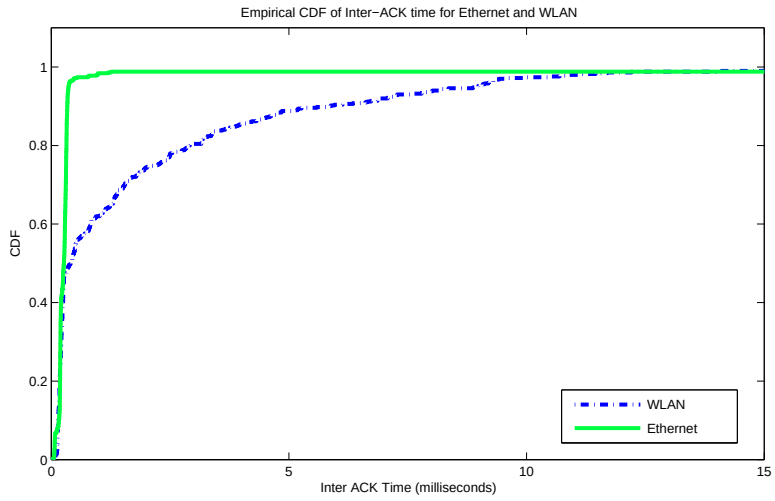


Figure 5.4: Inter-ACK time distributions for Ethernet and WLAN

5.2 Wireless Traffic Analysis and Results

For wireless side experiments, we have implemented the testbed as shown in Figure 5.5 for capturing the beacon frames. We have use Linksys WPA 55AG wireless card to capture the wireless traffic as this card supports the monitor mode and also because its driver provides open source code [45]. In this setup, we have used wireless card in monitor mode and captured

IEEE 802.11 traffic using tcpdump/wireshark [48] and kismet [49] which are installed on fingerprinter (as laptop in the Figure 5.5). We have captured beacon frames in various scenarios in pcap files, as discussed later.

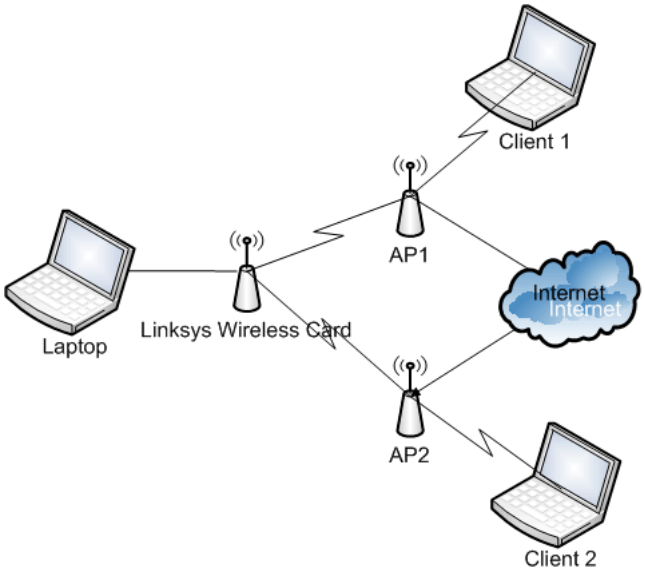


Figure 5.5: Experiment Testbed

For calculating clock skew (see Section 4.2), we need to know when a beacon frame reaches the wireless LAN card of the fingerprinter. We could have used arrival time which is generated by tcpdump. However, this arrival time is not that accurate because it includes variable processing time of OS, interrupt latency [28]. Therefore, use of tcpdump timestamp (see Figure 5.6) is not suitable for our purpose. We have used *radiotap header* information included in Intel 2200 driver [45]. This driver allows additional radiotap monitoring headers to be added to frames arriving at the wireless card which has a 8 byte timestamp field as shown as *radiotap.mactime* in Figure 5.6. The resolution of this timestamp is in microseconds. So, when Intel 2200 driver receives a frame, the current value of the TSF timer of fingerprinter is stored in the field of the radiotap header. We use the measure *parts per million*, essentially $\mu s/s$, denoted as *ppm*, to quantify clock skew. Apart from *radiotap.mactime*, we also have extracted beacon frame TSF using field *wlan_mgt.fixed.timestamp* for calculation of clock

skew. Using these two fields value and LSF (See Section 4.2), we have calculated clock skew of the particular AP. For calculating clock skew, we have written custom C program to calculate offsets and LSF was performed using *R-software* [50]. In addition to these fields, we have extracted field *radiotap.dbm_antsignal* from pcap files for RSS values in dBm. All required fields from pcap file are extracted using perl script and shown in Figure 5.6.

```
▽ Frame 24 (199 bytes on wire, 199 bytes captured)
    Arrival Time: Mar 27, 2010 15:58:11.925999000
    [Time delta from previous captured frame: 0.098099000 seconds]
    [Time delta from previous displayed frame: 0.098099000 seconds]
    [Time since reference or first frame: 2.347101000 seconds]
    Frame Number: 24
    Frame Length: 199 bytes
    Capture Length: 199 bytes
    [Frame is marked: False]
    [Protocols in frame: radiotap:wlan]
▽ Radiotap Header v0, Length 32
    Header revision: 0
    Header pad: 0
    Header length: 32
  ▷ Present flags: 0x0000486f
    MAC timestamp: 546119054
  ▷ Flags: 0x10
    Data Rate: 1.0 Mb/s
    Channel frequency: 2462 [BG 11]
  ▷ Channel type: 802.11b (0x00a0)
    SSI Signal: -41 dBm
    SSI Noise: -94 dBm
    Antenna: 1
  ▷ RX flags: 0x0000
▷ IEEE 802.11 Beacon frame, Flags: ........C
▽ IEEE 802.11 wireless LAN management frame
  ▽ Fixed parameters (12 bytes)
      Timestamp: 0x00015C002AFE71B0
      Beacon Interval: 0.102400 [Seconds]
    ▷ Capability Information: 0x0431
  ▷ Tagged parameters (127 bytes)
```

Figure 5.6: Beacon Frame: Required Fields

To validate, how RSS plays an important role in MAC spoofing detection, we have plotted RSS values for the cases when (1) legitimate AP is transmitting alone and (2) rogue AP is also transmitting with the legitimate AP. As shown in Figure 5.7, when a single device is transmitting, its distribution is quite steady. However, when two devices are transmitting, then due to inherent differences of the physical locations, their RSS distributions are quite different and hence fluctuate many times. This leads to detection of rogue AP having same MAC address. However, this scheme does not work, when only rogue AP is transmitting.

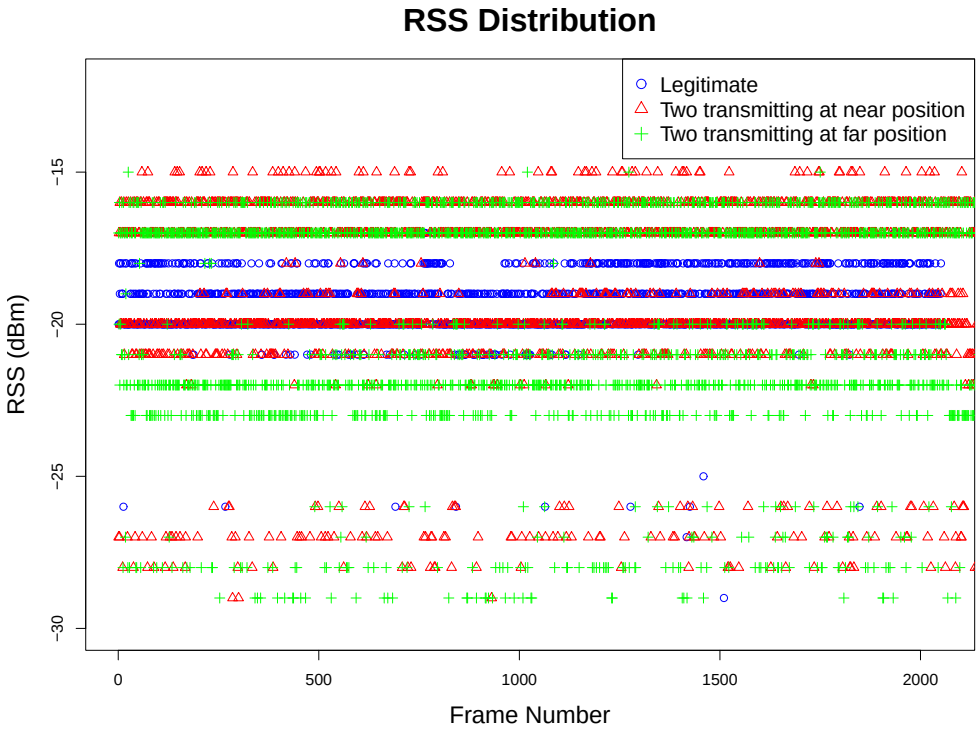


Figure 5.7: RSS distribution with and without MAC Spoofing

In order to test stability of clock skew, we have performed an experiment in different scenarios using Belkin wireless router. To do this test, we have calculated clock skew of Belkin router for five measurements and it came pretty consistent as shown in Table **??**. After that, we have switched off device for an hour and then restarted again. Then we calculated clock skew of device for five measurements and we see, it is quite consistent with previous value. We have performed same test next day to find out whether clock skew remains constant as mentioned before. In all cases, we have found consistency in clock skew of device as shown in Figure 5.8 and hence can be used as fingerprint. However, it fluctuates with temperature and can be manipulated as proven by authors of [39].

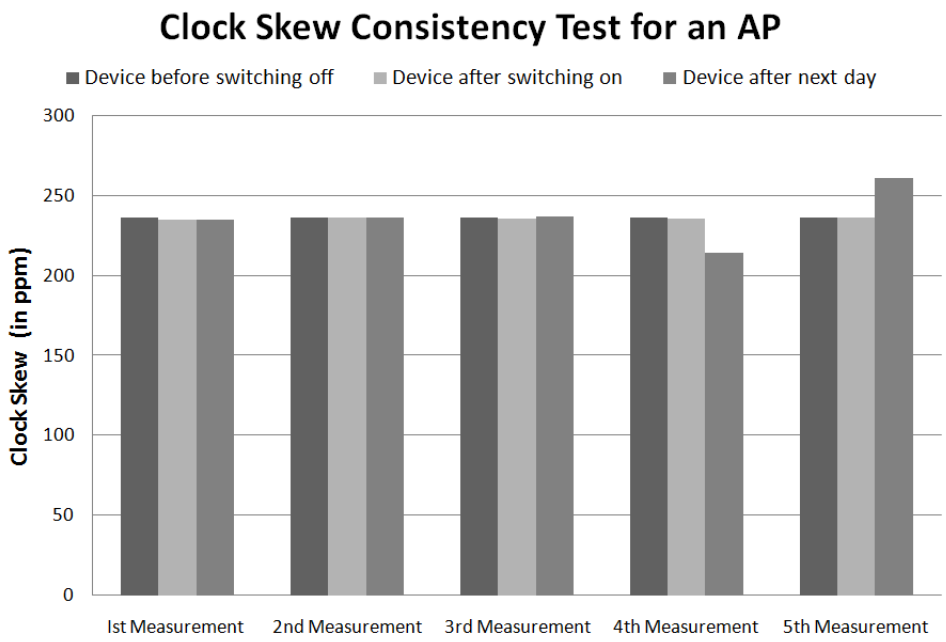


Figure 5.8: Clock Skew Consistency Test For An AP

The objectives of our experiments are to determine whether our fingerprinting technique is able to (1) accurately differentiate between unique devices (2) produce consistent results for each device over time, on separate occasions and under varying networks loads and (3) distinguish between different machines in the most challenging case where the machines have the same specifications (vendor model, RAM, processor etc), and equipped with the same OS and wireless NIC drivers. From henceforth, we will call these devices *similar type devices* and other types of devices as *different type devices*. To test the classifier performance, we have used N-Fold Cross Validation (In our case we have used $N = 10$). In this approach, we randomly partition dataset into N sets of equal size and run the learning algorithm N times. Each time, one of the N sets is the test set, and the model is trained on the remaining $N - 1$ sets. Performance of any classifier is given by the following parameters and they are later used in performance tables for various experiments.

- Confusion Matrix or Contingency Table: It is a table with two rows and two columns

that reports the number of True Positives (TP), False Positives (FP), False Negatives (FN) and True Negatives (TN) as shown in Table 5.1.

Table 5.1: Confusion Matrix

		Actual Value	
		True	False
Prediction Outcome	True	True Positive(TP)	False Positive(FP)
	False	False Negative(FN)	True Negative(TN)

- True Positive Rate(TPR) or Sensitivity: This measures the proportion of actual positives which are correctly identified as such.

$$TPR = \frac{TP}{TP + FN} \tag{5.1}$$

- False Positive Rate or Fall Out: This measures the proportion of non-relevant documents that are retrieved, out of all non-relevant documents available.

$$FPR = \frac{FP}{FP + TN} \tag{5.2}$$

- Precision: It is the fraction of the documents retrieved that are relevant to the user's information need. In binary classification, precision is analogous to positive predictive value. Precision takes all retrieved documents into account.

- Recall: It is the fraction of the documents that are relevant to the query that are successfully retrieved. In binary classification, recall is called sensitivity.

- F-Measure: It is a measure of a test's accuracy. It considers both the precision and the recall to the test to compute the score. It is defined as the harmonic mean of precision and recall.

- Receiver Operating Characteristic (ROC): It is a graphical plot of TP vs. FP or TPR vs. FPR, used to select possibly optimal models.

The first set of experiments was conducted with two APs (Linksys G Routers) with identical specifications. The experiment involved collecting wireless traffic for five hours. We have conducted experiments in the following scenarios:

1. One AP transmitting at a time, it means that at any instant only one device is transmitting.
2. Both devices are transmitting at same time when they are placed quite near (around one foot).
3. Both devices are transmitting at same time when they are placed at far positions (around fifteen feet).

After that we have calculated clock skews of both devices in each scenario. So when both devices are transmitting at the same time, their frames are interspersed and so when calculating clock skew using LSF, its value is exceptionally larger than the actual clock skews of each of the contributing AP as shown in Table 5.2.

Table 5.2: Clock Skew Calculations using Linear Regression

	Linksys1 (in ppm)	Linksys2 (in ppm)	Both Transmitting together at near position (in ppm)	Both Transmitting together at far position (in ppm)
1st Measurement	242.6352	240.1069	-26032.77	28882.45
2nd Measurement	242.0631	237.8908	-9428.759	19941.61
3rd Measurement	241.6687	236.6897	-9671.872	28657.23
4th Measurement	241.2772	235.8663	-5666.581	22599.83
5th Measurement	241.0768	235.2793	10249.15	-7890.155

After calculating clock skews in each scenarios, we have used clock skew and RSS combination to train naïve Bayes classifier to classify between two similar type of devices. We have used version 3.6 of the *WEKA* software suite [51], which supports naïve Bayes classifier. WEKA, written in the Java language, allows tracking of its memory footprint. We have used 10–fold cross validation (stratified). Results are summarized in following tables for each scenario:

Case 1 When both devices are transmitting at different time. In this case, 98.9123% accuracy is achieved in differentiating the similar devices. We see some false negatives because of RSS distributions of devices are quite close and clock skews are nearly in same order. Results are summarized in Tables 5.3, 5.4 and 5.5.

Table 5.3: Attribute Summary

Attribute	Class	
	Yes	No
Clockskew		
mean	241.7639	237.3542
std. dev.	0.61	1.6815
RSS		
mean	-37.7677	-36.4132
std. dev.	4.7465	1.5489

Table 5.4: Confusion Matrix

		Actual Value	
		True	False
Prediction Outcome	True	10817	0
	False	236	10645

Table 5.5: Detailed Accuracy By Class

Total Number of Instances: 21698							
	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0.022	0.979	1	0.989	0.998	yes
	0.978	0	1	0.978	0.989	0.998	no
Weighted Avg.	0.989	0.011	0.989	0.989	0.989	0.998	
Accuracy = 98.9123%							

Case 2 When both devices are transmitting simultaneously at near by location. For this case, we also get some false negatives because of the similar RSS distributions. However, we get a smaller number of false negatives than case 1 because the difference between clock skews is quite high in this case. This is caused by interspersing of frames of both devices. In this case, 99.7852% accuracy is achieved in differentiating the similar devices. Results are summarized in Tables 5.6, 5.7 and 5.8.

Table 5.6: Attribute Summary

Attribute	Class	
	Yes	No
Clockskew		
mean	0	-6463.0963
std. dev.	671.8874	11525.045
RSS		
mean	-37.8376	-36.8478
std. dev.	4.7046	2.769

Table 5.7: Confusion Matrix

		Actual Value	
		True	False
Prediction Outcome	True	10817	0
	False	47	11012

Table 5.8: Detailed Accuracy By Class

Total Number of Instances: 21876							
	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0.004	0.996	1	0.998	1	yes
	0.996	0	1	0.996	0.998	1	no
Weighted Avg.	0.998	0.002	0.998	0.998	0.998	1	
Accuracy = 99.7852%							

Case 3 When both devices are transmitting simultaneously at far location. In this case, 100% accuracy is achieved in differentiating the similar devices. This is due to large difference in clock skews and RSS distributions. Results are summarized in Tables 5.9, 5.10 and 5.11.

Table 5.9: Attribute Summary

	Class	
Attribute	Yes	No
Clockskew		
mean	0	18713.4857
std. dev.	680.9742	13867.9098
RSS		
mean	-37.7677	-41.2513
std. dev.	4.7465	3.269

Table 5.10: Confusion Matrix

		Actual Value	
		True	False
Prediction Outcome	True	10817	0
	False	0	11133

Table 5.11: Detailed Accuracy By Class

Total Number of Instances: 21950							
	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	1	yes
	1	0	1	1	1	1	no
Weighted Avg.	1	0	1	1	1	1	
Accuracy = 100%							

Similarly, we have simulated same scenarios when both devices are different in configuration. We have used Linksys and Netgear wireless routers for this purpose. All the scenarios and calculations are same as described previously in the case of similar devices. Results are summarized in following tables. Clock skews calculations using LSF are shown in Table 5.12.

Table 5.12: Clock Skew Calculations using Linear Regression

	Linksys (in ppm)	Netgear (in ppm)	Both Transmitting together at near position (in ppm)	Both Transmitting together at far position (in ppm)
1st Measurement	-7.386643	-15.91502	-121842.7	+17775.67
2nd Measurement	-7.381594	-16.21453	-36013.98	+22089.00
3rd Measurement	-7.429683	-16.34325	+58023.86	-25353.55
4th Measurement	-7.295461	-16.31122	+41420.07	+12155.64
5th Measurement	-7.290116	-16.46016	+75193.15	+19551.48

Case 1 When both devices are transmitting at different time. In this case, 100% accuracy is achieved in differentiating the different devices. Results are summarized in Tables 5.13, 5.14 and 5.15.

Table 5.13: Attribute Summary

Attribute	Class	
	Yes	No
Clockskew		
mean	-7.1323	-16.3023
std. dev.	0.1698	0.1698
RSS		
mean	-18.8364	-18.8764
std. dev.	1.0956	2.9469

Table 5.14: Confusion Matrix

		Actual Value	
		True	False
Prediction Outcome	True	10468	0
	False	0	10458

Table 5.15: Detailed Accuracy By Class

Total Number of Instances: 20926							
	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	1	yes
	1	0	1	1	1	1	no
Weighted Avg.	1	0	1	1	1	1	
Accuracy = 100%							

Case 2 When both devices are transmitting simultaneously at near location. In this case, 99.6904% accuracy is achieved in differentiating the different devices. Results are summarized in Tables 5.16, 5.17 and 5.18.

Table 5.16: Attribute Summary

Attribute	Class	
	Yes	No
Clockskew		
mean	0	262.2626
std. dev.	3648.812	77188.8441
RSS		
mean	-18.9922	-18.8412
std. dev.	0.9699	3.2123

Table 5.17: Confusion Matrix

		Actual Value	
		True	False
Prediction Outcome	True	10402	66
	False	0	10852

Table 5.18: Detailed Accuracy By Class

Total Number of Instances: 21320							
	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.994	0	1	0.994	0.997	1	yes
	1	0.006	0.994	1	0.997	1	no
Weighted Avg.	0.997	0.003	0.997	0.997	0.997	1	
Accuracy = 99.6904%							

Case 3 When both devices are transmitting simultaneously at far location. In this case, 99.7094% accuracy is achieved in differentiating the different devices. Results are summarized in Tables 5.19, 5.20 and 5.21.

Table 5.19: Attribute Summary

Attribute	Class	
	Yes	No
Clockskew		
mean	0	8573.2942
std. dev.	878.5657	17775.6189
RSS		
mean	-19.0552	-20.1968
std. dev.	1.1609	3.9591

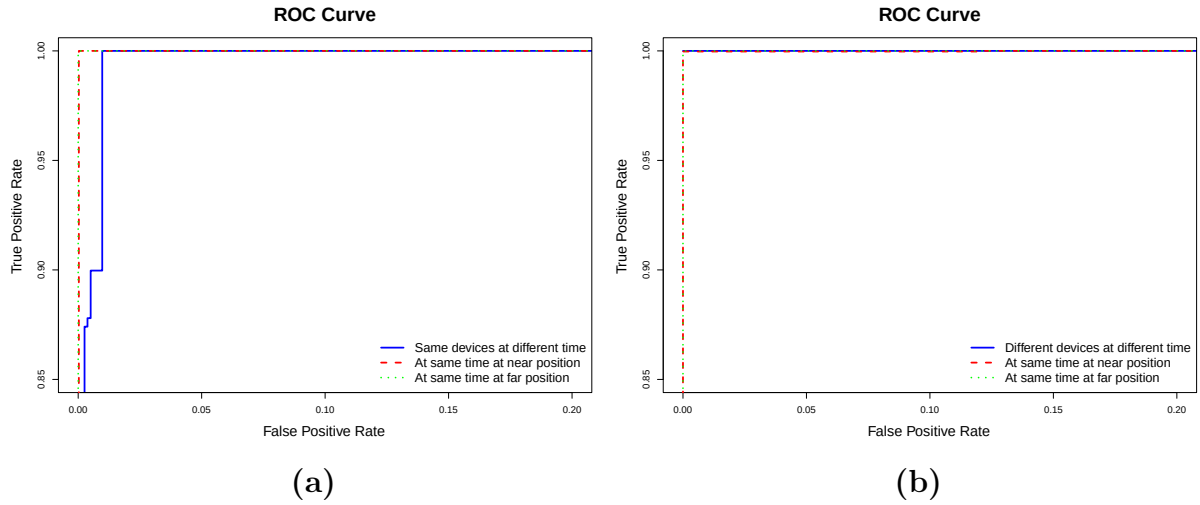
Table 5.20: Confusion Matrix

		Actual Value	
		True	False
Prediction Outcome	True	10406	62
	False	0	10864

Table 5.21: Detailed Accuracy By Class

Total Number of Instances: 21332							
	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.994	0	1	0.994	0.997	1	yes
	1	0.006	0.994s	1	0.997	1	no
Weighted Avg.	0.997	0.003	0.997	0.997	0.997	1	
Accuracy = 99.7094%							

Figure 5.9 shows the ROC curves for similar and different types of devices.



(a) **(b)**

Figure 5.9: ROC Curve for (a) Similar type devices and (b) Different type devices

From above results, we deduce that on an average, our fingerprinting technique is 98% accurate at differentiating between similar types of devices in various scenarios. It has an accuracy of 99.5% in correctly identifying different type of devices.

Chapter 6

Conclusions and Future Work

In this chapter we present concluding remarks about the work presented in this thesis. In Section 6.2, we will discuss about the future work.

6.1 Concluding Remarks

In this thesis, we have tackled one of the most common security problem in IEEE 802.11 WLAN networks which is presence of *rogue AP*. This problem still exists even with current security mechanisms provided by 802.11. For detection of rogue AP, we proposed a suite of lightweight security solutions for wireless networks that complement the conventional authentication services. We have developed novel hybrid framework for detecting rogue AP by applying statistical analysis of all data from both wired scans and wireless surveillance.

For wired side approach, we have developed an algorithm based on weighted sequential hypothesis test. This algorithm takes input as TCP-ACK pairs and exploits the fundamental properties of the 802.11 CSMA/CA MAC protocol and the half duplex nature of wireless channels to differentiate Ethernet and WLAN TCP traffic. We have presented offline detector module which parses pcap file and using Algorithm 3.1, it differentiates between Ethernet

and WLAN hosts. This detector module does this by taking care of OS diversity and latency. Extensive experiments in various scenarios and over hosts with various operating systems have demonstrated the excellent performance of our detector module. Results are summarized in Section 5.1. The detector module provides rapid detection and is extremely accurate in detecting Ethernet and WLAN hosts.

From WLAN side, we proposed a new wireless fingerprinting that differentiates between unique devices through timing analysis of 802.11 beacon frames. This fingerprinting technique uses combination of clock skew and RSS. The effectiveness of this mechanism is measured through naïve Bayes classifier. The technique is also involuntary, meaning the method can not be circumvented, saved by physical alteration of the AP. As well as being an effective technique, its implementation is very simple passive measurement with minimum hardware requirements. Almost any 802.11 card could be used for that. This technique implies taking an step forward towards the creation of valid profiles that will allow us to detect anomalies in WLANs. We evaluated this technique using traces from various scenarios and setups as discussed in Section 5.2. Our exploration results indicate that the use of this fingerprinting technique appears to be an efficient and robust method for detecting MAC spoofing in WLANs. The naïve Bayes classifier was able to differentiate the APs with the average success rate of 99%.

We conclude by stating that this hybrid framework can be very efficient and cost-effective alternative to detect the presence of rogue AP at any hot-spots or enterprise networks and can be used with existing security mechanisms for WLANs.

6.2 Directions for Future Work

In this section, we highlight and discuss some issues related to the work presented in this thesis. We would address these issues in our future work.

The performance comparison between weighted sequential hypothesis test and sequential

hypothesis test can be done to have better understanding of convergence of both tests under different network setups and various traffic scenarios. Detector module (see Section 3.4) focuses on TCP traffic, which is about the most of Internet traffic. This approach can be extended by taking account of *User Datagram Protocol (UDP)* which does not have feedback policy. Additionally, the detector module is placed one hop from the switch; future plans can include to detect the wireless hosts multiple hops downstream.

This hybrid framework has been tested thoroughly for infrastructure-based architecture. This framework could effectively be implemented in ad-hoc networks as they also use management frames that can suffer from the same kind of attacks. In infrastructure architecture, AP is stationary and hence its RSS fingerprints are quite stable. However, in ad-hoc networks, nodes are mobile. A key element of future work is to adapt these mechanisms for mobile nodes. For wireless traffic analysis, current fingerprinting approach takes clock skew and RSS as features. In future, this technique can add more feature metrics as sequence number tracking and NIC's driver fingerprinting. This will in turn increase the robustness of the technique. Within the work detailed in Section 5.2, this thesis uses naïve Bayes classifier to classify the devices. Other machine-learning techniques can be explored for classification. In future, this framework can be tested for spatial independence through the models trained using one set of network-traces upon an entirely different location.

Bibliography

[1] "Wireless lan medium access control (mac) and physical layer (phy) specifications." IEEE Standard 802.11-2007, June 2007. `http://standards.ieee.org/getieee802/802.11.html`.

[2] W. Wei, K. Suh, B. Wang, Y. Gu, J. Kurose, D. Towsley, and S. Jaiswal, "Passive online detection of 802.11 traffic using sequential hypothesis testing with tcp ack-pairs," vol. 8, (Los Alamitos, CA, USA), pp. 398–412, IEEE Computer Society, 2008.

[3] J. Schiller, *Mobile Communications*. Addison Wesley, 2nd ed., September 2003.

[4] T. Karygianni and L. Owens, "Wireless network security - 802.11, bluetooth and handheld devices." National Institute for Standards and Technology, USA, November 2002.

[5] E. B. O. L. Frankel, S. and K. Scarfone, "Establishing wireless robust security networks: A guide to ieee 802.11i," Feb 2007. `http://csrc.nist.gov/publications/nistpubs/800-97/SP800-97.pdf`.

[6] L. Ma, A. Y. Teymorian, X. Cheng, and M. Song, "Rap: protecting commodity wi-fi networks from rogue access points," in *QSHINE '07: The Fourth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness; Workshops*, (New York, NY, USA), pp. 1–7, ACM, 2007.

[7] J. Yeo, M. Youssef, and A. Agrawala, "A framework for wireless lan monitoring and its applications," in *WiSe '04: Proceedings of the 3rd ACM workshop on Wireless security*, (New York, NY, USA), pp. 70–79, ACM, 2004.

[8] "Airmagnet." `http://www.airmagnet.com/`.

[9] "Netstumbler." `http://www.netstumbler.com/`.

[10] "Arubanetworks." `http://www.arubanetworks.com/`.

[11] "Airdefence." `http://airdefense.net/`.

[12] "Airwave." `http://www.airwave.com/`.

[13] "Proxim." `http://www.proxim.com/`.

[14] "Cisco wireless lan solution engine (wlse)." `http://www.cisco.com/`.

[15] P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill, "Enhancing the security of corporate wi-fi networks using dair," in *MobiSys '06: Proceedings of the 4th international conference on Mobile systems, applications and services*, (New York, NY, USA), pp. 1–14, ACM, 2006.

[16] A. Adya, P. Bahl, R. Chandra, and L. Qiu, "Architecture and techniques for diagnosing faults in ieee 802.11 infrastructure networks," in *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 30–44, ACM, 2004.

[17] M. K. Chirumamilla and B. Ramamurthy, "Agent based intrusion detection and response system for wireless lans," in *in ICC 03, IEEE International Conference on Communications*, pp. 492–496, 2003.

[18] G. Joshua Wright and C. Joshua, "Detecting Wireless LAN MAC Address Spoofing," *Cisco Certified Network Associate*, 2003.

[19] Q. Li and W. Trappe, "Light-weight detection of spoofing attacks in wireless networks," *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, pp. 845–851, 2006.

[20] Y. Chen and R. P. Martin, "Detecting and localizing wireless spoofing attacks," in *In Proceedings of the 4th Annual IEEE Conference on Sensor Mesh and Ad Hoc Communications and Networks*, 2007.

[21] J. Hall, M. Barbeau, and E. Kranakis, "Detection of transient in radio frequency fingerprinting using signal phase," *Wireless and Optical Communications*, pp. 13–18, 2003.

[22] C. Corbett, R. Beyah, and J. Copeland, "A passive approach to wireless NIC identification," in *IEEE International Conference on Communications, 2006. ICC'06*, vol. 5, 2006.

[23] D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints," in *WiSe '06: Proceedings of the 5th ACM workshop on Wireless security*, (New York, NY, USA), pp. 43–52, ACM, 2006.

[24] O. Ureten and N. Serinken, "Wireless security through RF fingerprinting," *Electrical and Computer Engineering, Canadian Journal of*, vol. 32, no. 1, pp. 27–33, 2007.

[25] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. Randwyk, and D. Sicker, "Passive data link layer 802.11 wireless device driver fingerprinting," in *Proc. USENIX Security Symposium*, pp. 167–178, USENIX Association, 2006.

[26] J. Cache, "Fingerprinting 802.11 implementations via statistical analysis of the duration field," *Uninformed. org*, vol. 5, 2006.

[27] T. Kohno, A. Broido, and K. Claffy, "Remote physical device fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, pp. 93–108, 2005.

[28] S. Jana and S. K. Kasera, "On fast and accurate detection of unauthorized wireless access points using clock skews," in *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*, (New York, NY, USA), pp. 104–115, ACM, 2008.

[29] C. D. Mano, A. Blaich, Q. Liao, Y. Jiang, D. A. Cieslak, D. C. Salyers, and A. Striegel, "Ripps: Rogue identifying packet payload slicer detecting unauthorized wireless hosts through network traffic conditioning," *ACM Trans. Inf. Syst. Secur.*, vol. 11, no. 2, pp. 1–23, 2008.

[30] G. Y. B. S. R. Beyah, S. Kangude and J. Copeland, "Rogue access point detection using temporal traffic characteristics," in *In Proceedings of the IEEE GLOBECOM*, December 2004.

[31] M. S. S. Shetty and L. Ma, "Rogue access point detection by analyzing network traffic characteristics," in *In MILCOM*, October 2007.

[32] V. Baiamonte, K. Papagiannaki, G. Iannaccone, and P. D. Torino, "Detecting 802.11 wireless hosts from remote passive observations," in *In Proc. IFIP/TC6 Networking*, 2007.

[33] H. Yin, G. Chen, and J. Wang, "Detecting protected layer-3 rogue aps," in *In Proceedings of the Fourth IEEE International Conference on Broadband Communications, Networks, and Systems*, 2007.

[34] L. Ma, A. Teymorian, and X. Cheng, "A hybrid rogue access point protection framework for commodity wi-fi networks," pp. 1220 –1228, April 2008.

[35] "Nmap." `http://www.nmap.org/`.

[36] A. Wald, *Sequential analysis.* J. Wiley and Sons, 1947.

[37] V. Paxson, "On calibrating measurements of packet transit times," in *Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, p. 21, ACM, 1998.

[38] S. Moon, P. Skelly, and D. Towsley, "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*, vol. 1, 1999.

[39] C. Arackaparambil, S. Bratus, A. Shubina, and D. Kotz, "On the Reliability of Wireless Fingerprinting using Clock Skews," tech. rep., Technical Report TR2010-661, Dartmouth College, Computer Science, Hanover, NH, 2010.

[40] T. Rappaport *et al.*, *Wireless communications: principles and practice.* Prentice Hall PTR New Jersey, 2002.

[41] D. Madory, *New Methods of Spoof Detection in 802.11 b Wireless Networking.* PhD thesis, Citeseer, 2006.

[42] A. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, p. 60, ACM, 2005.

[43] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall, "802.11 user fingerprinting," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, p. 110, ACM, 2007.

[44] I. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann Pub, 2005.

[45] "Intel 2200 driver, intel 2200 firmware and radiotap header, website 2006." `http://ipw2200.sourceforge.net/`.

[46] V. Jacobson, C. Leres, and S. McCanne, "The tcpdump manual page," *Lawrence Berkeley Laboratory, Berkeley, CA*, 1989.

[47] M. Guide, "The MathWorks," *Inc., Natick, MA*, vol. 5, 1998.

[48] G. Combs *et al.*, "Wireshark," *Web page: http://www.wireshark.org/*, 2007.

[49] M. Kershaw, "Kismet wireless, http://www. kismetwireless. net," *Retrieved from the Web*, 2007.

[50] R. Ihaka and R. Gentleman, "R: A language for data analysis and graphics," *Journal of computational and graphical statistics*, vol. 5, no. 3, pp. 299–314, 1996.

[51] S. R. Garner, "Weka: The waikato environment for knowledge analysis," in *In Proc. of the New Zealand Computer Science Research Students Conference*, pp. 57–64, 1995.