Departament d'Enginyeria Telemàtica
Universitat Politècnica de Catalunya

Ph.D. Thesis

Contributions to the Security of Cognitive Radio Networks

Ph.D. Candidate: Olga León Abarca
Ph.D. Advisor: Juan Hernández Serrano

January 2012

A mis padres.
A mi hermano David.
A mis tías Asun y Pili.
A mis abuelos María y Cesáreo.

Abstract

The increasing emergence of wireless applications along with the static spectrum allocation followed by regulatory bodies has led to a high inefficiency in spectrum usage, and the lack of spectrum for new services. In this context, Cognitive Radio (CR) technology has been proposed as a possible solution to reuse the spectrum being underutilized by licensed services. CRs are intelligent devices capable of sensing the medium and identifying those portions of the spectrum being unused. Based on their current perception of the environment and on that learned from past experiences, they can optimally tune themselves with regard to parameters such as frequency, coding and modulation, among others. Due to such properties, Cognitive Radio Networks (CRNs) can act as secondary users of the spectrum left unused by their legal owners or primary users, under the requirement of not interfering primary communications.

The successful deployment of these networks relies on the proper design of mechanisms in order to efficiently detect spectrum holes, adapt to changing environment conditions and manage the available spectrum. Furthermore, the need for addressing security issues is evidenced by two facts. First, as for any other type of wireless network, the air is used as communications medium and can easily be accessed by attackers. On the other hand, the particular attributes of CRNs offer new opportunities to malicious users, ranging from providing wrong information on the radio environment to disrupting the cognitive mechanisms, which could severely undermine the operation of these networks.

In this PhD thesis we have approached the challenge of securing Cognitive Radio Networks. Because CR technology is still evolving, to achieve this goal involves not only providing countermeasures for existing attacks but also to identify new potential threats and evaluate their impact on CRNs performance.

The main contributions of this thesis can be summarized as follows. First, a critical study on the State of the Art in this area is presented. A qualitative analysis of those threats to CRNs already identified in the literature is provided, and the efficacy of existing countermeasures is discussed. Based on this work, a set of guidelines are defined in order to design a detection system for the main threats to CRNs. Besides, a high level description of the components of this system is provided, being it the second contribution of this thesis.

The third contribution is the proposal of a new cross-layer attack to the Transmission Control Protocol (TCP) in CRNs. An analytical model of the impact of this attack on the throughput of TCP connections is derived, and a set of countermeasures in order to detect and mitigate the effect of such attack are proposed.

One of the main threats to CRNs is the Primary User Emulation (PUE) attack. This attack prevents CRNs from using available portions of the spectrum and can even lead to a Denial of Service (DoS). In the fourth contribution of this thesis, a cooperative location method is proposed in order to deal with such attack. The method relies on a set of time measures provided by the members of the network and allows estimating the position of an emitter. This estimation is then used to determine the legitimacy of a given transmission and detect PUE attacks.

Cooperative methods are prone to be disrupted by malicious nodes reporting false data. This problem is addressed, in the context of cooperative location, in the fifth and last contribution of this thesis. A method based on Least Median Squares (LMS) fitting is proposed in order to detect forged measures and make the location process robust to them.

The efficiency and accuracy of the proposed methodologies are demonstrated by means of simulation.

Resumen

La aparición de nuevas aplicaciones inalámbricas, junto con las políticas estáticas de asignación de espectro aplicadas por las agencias reguladoras, ha conllevado un uso muy ineficiente del espectro y la escasez de bandas de frecuencia para servicios emergentes. En este contexto, la tecnología de Radio Cognitiva se ha propuesto como solución para reutilizar aquellas porciones del espectro licenciado en desuso por parte de sus propietarios legítimos. Las Radios Cognitivas son dispositivos inteligentes capaces de escuchar el medio e identificar las bandas de frecuencia libres. Basándose en su percepción del entorno y en experiencias previas, estos dispositivos pueden autoconfigurarse de forma óptima en cuanto a parámetros de transmisión se refiere, como por ejemplo frecuencia, codificación o modulación. Como consecuencia, las Redes de Radios Cognitivas pueden actuar como usuarios secundarios del espectro liberado por los usuarios primarios o legítimos del mismo, siempre y cuando no interfieran las comunicaciones primarias.

El despliegue con éxito de Redes de Radio Cognitivas depende en gran medida de la implementación de mecanismos que permitan, de un modo eficiente, detectar qué frecuencias no están siendo utilizadas por usuarios primarios, adaptarse a los cambios en el entorno y gestionar el espectro disponible. Además, la necesidad de tratar aspectos de seguridad en dichas redes queda evidenciada principalmente por dos hechos. En primer lugar, tal y como ocurre en todo tipo de redes inalámbricas, un atacante tiene acceso directo al medio. Por otro lado, las peculiares características de estas redes ofrecen nuevas oportunidades a usuarios maliciosos de deteriorar el funcionamiento de las mismas, por ejemplo, proporcionando información falsa sobre el medio o alterando el funcionamiento de los mecanismos cognitivos.

En esta tesis se aborda el reto de proporcionar seguridad a Redes de Radios Cognitivas. Puesto que se trata de una tecnología todavía en desarrollo, cumplir con este objetivo implica no sólo proporcionar contramedidas para los ataques que ya han sido presentados en la literatura, sino identificar nuevas amenazas y evaluar el impacto que pueden tener en el funcionamiento de dichas redes.

Las principales contribuciones de esta tesis se pueden resumir tal y como se detalla a continuación. En primer lugar, se elabora un estudio crítico del estado del Arte en este campo. Se proporciona un anális de las amenazas identificadas en la literatura y se discute la eficacia de las correspondientes contramedidas propuestas hasta el momento. A raíz de este trabajo, se proporcionan un conjunto de pautas a seguir para el diseño de un sistema de detección de ataques para Cognitive Radio Networks (CRNs). Se definen los componentes de dicho sistema y se describen las principales características de cada uno de ellos. Esta descripción a alto nivel constituye la segunda contribución de esta tesis.

La tercera contribución es la propuesta de un nuevo ataque cross-layer al protocolo Transmission Control Protocol (TCP). Se proporciona un modelo analítico del impacto que dicho ataque puede tener sobre el caudal de las conexiones TCP que se establecen en Redes de Radio Cognitivas, así como un conjunto de contramedidas para detectar y mitigar el efecto del mismo.

Los ataques de emulación de usuarios primarios constituyen una de las mayores amenazas a Redes de Radio Cognitivas. En la cuarta contribución de esta tesis, se propone un método cooperativo de localización para tratar dichos ataques. Dicho método permite estimar la posición de un emisor basándose en un conjunto de medidas temporales que proporcionan los miembros de la red. A partir de esta estimación, puede determinarse la legitimidad de una determinada transmisión y detectar ataques de emulación de primario.

Los mecanismos cooperativos pueden dar rienda suelta a comportamientos maliciosos en los que uno o varios miembros de la red proporcionan información falsa con el objetivo de degradar el funcionamiento de la red. La quinta y última contribución de esta tesis aborda este problema en el contexto de un método cooperativo de localización. Se propone un mecanismo basado en la mediana de mínimos cuadrados que permite detectar medidas falsas y proporcionar robustez al proceso de localización frente a las mismas.

La eficiencia y la precisión de las diversas metodologías propuestas se demuestra mediante simulación.

Agradecimientos

Con estas líneas quiero expresar mi agradecimiento a todas aquellas personas que de un modo u otro han hecho posible esta tesis.

A mi director Juan, por implicarse completamente en este trabajo. Muchísimas gracias por todas las horas que me has dedicado, por todos los consejos y por ser un amigo, además de un buen director.

Al Miquel, per haver-me acollit dins el grup ISG, per totes les aportacions que ha fet a aquesta tesi i pel tracte exquisit que m'ha donat durant tot aquest temps.

A la resta de membres del grup d'ISG. A Óscar, por ser un truhán (en su segunda acepción) y un señor a la vez.

Als meus companys de la secció de Castelldefels. Al Javi, per amenitzar tots els dinars substituint les discussions sobre bits i xarxes per interessants converses sobre literatura, cine i sexe.

A Cristina, Lourdes y Mariajo, las tres "señus" que conocí en la residencia cuando empecé a estudiar en la universidad. ¡Gracias por confiar en mí y por haberme acompañado durante todo este tiempo!

Al Pau, el Xavi i la Divi, per tots els dijous que m'han fet oblidar que sóc una adulta responsable. I per la resta de dies en que sempre han estat disposats a escoltar-me. Moltes gràcies Pau, per tots els consells linuxeros i per fer-me saber que puc comptar amb tu.

A Enrica, por todo lo que hemos compartido dentro y fuera de la universidad, por ayudarme a superar los malos momentos y a disfrutar de los buenos, y por esa complicidad que no es fácil encontrar. Grazie tante, Enri!

Als meus amics de Lleida, en especial a l'Imma, el Miquel, la Montse i el Pedro, que m'han ajudat a desconnectar del món universitari i del ritme frenètic de Barcelona. I molt especialment a la Montse. Gràcies per ser com ets, per estar al meu cantó sempre que ho he necessitat i per les innumerables nits en que m'has acollit a casa teva i m'has contagiat la teva alegria.

Gracias a Susana G., Mónica, Susana B., Lorenza, Josefina... y a todos aquellos que estando en Lleida, Barcelona, Zaragoza o Sebastopol, me habéis hecho sonreír.

Y por supuesto, mi mayor agradecimiento va dirigido a mi familia, mi piedra angular. A mis padres Fernando y Mara, por su apoyo incondicional, por ser tan pacientes conmigo no sólo durante el desarrollo de esta tesis sino durante toda mi existencia y por los valores que me han inculcado. Muchísimas gracias, moltíssimes gràcies per tot el que heu fet per mi! A mi hermano David, el leoncito de la familia, por haber crecido conmigo y por la ternura que despierta en mí. A las tres mamás extra que la vida me ha dado: mi yaya María y mis dos tatas Pili y Asun, por su cariño y por darlo todo y mucho más por mí. A mi yayo Cesáreo, que siempre fue un ejemplo a seguir para mí. Als meus tiets Àngel i Maite, per tenir una visió tan positiva de les coses i pel seu ímpetu en mantenir unida a tota la família.

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Introduction

With the rapid increase of wireless applications, spectrum is becoming a scarce resource since all frequencies below 3 GHz have been allocated to specific services. Regulatory agencies, such as the Federal Communications Commission (FCC), allocate spectrum for particular types of services that are then licensed to bidders for a fee. These allocations and licenses are static in nature leading to considerable inefficiency in spectrum utilization, and generating an unnecessary shortage of spectrum. In addition, actual measurements [McHenry, 2005] show that most of the allocated spectrum is vastly underutilized at any specific location and time, as shown in Figure 1.1. These facts have stimulated a wide range of activities and regulation communities in searching for better spectrum management policies, such as the upcoming of the Dynamic Spectrum Access (DSA) paradigm. DSA proposes a new spectrum management scheme where wireless networks are allowed to opportunistically access the spectrum, either left unused by primary systems, i.e., the owners of spectrum rights, or unlicensed spectrum that must be shared among different networks.

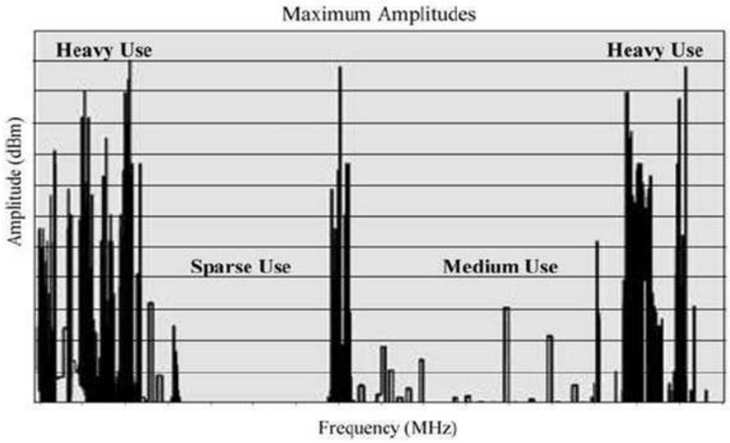


Figure 1.1: Spectrum usage [1]

[1] Image taken from [Akyildiz et al., 2006]

Conventional radios can only access one area of the radio spectrum, which is selected at the time of design. Because of this, they do not meet the adaptability requirements of DSA scenarios. The emergence of Software Defined Radios (SDRs) was the first step to make DSA networks feasible. A SDR can be defined as a radio where transmit frequencies, modulation type and other Radio Frequency (RF) parameters can be configured and reconfigured by software. As a consequence, they can provide a wide range of services with variable Quality of Service (QoS) in order to adapt to different network technologies and to the dynamics of radio propagation.

The addition of cognition capability to a SDR lead to the idea of a Cognitive Radio (CR), an intelligent radio capable of tuning itself based on its own perception of the spectrum availability and the environment conditions. Recent improvements in computational abilities of current electronic devices and in artificial intelligence have led to consider CR technology as a feasible solution to the overcrowding of the spectrum space. Besides, the Notice of Proposed Rule Making (NPRM) issued by the FCC [FCC, 2003b] in 2003 aroused special interest regarding Cognitive Radio Networks (CRNs). This NPRM suggests the use of CR technology to increase spectrum efficiency by allowing unlicensed devices to operate at times or in locations where licensed spectrum is not in use, such as TV bands in rural areas. In this context, a CRN could act as secondary user of the spectrum whenever primary systems are not using it and without causing harmful interference to them.

The successful deployment of CRNs relies on the design of several mechanisms, such as spectrum sensing in order to detect spectrum holes and ensure non-interfering coexistence with primary users, or spectrum management schemes to fairly distribute the available bandwidth among secondary users. Unfortunately, these particular attributes offer new unique opportunities to malicious attackers willing to disrupt the operation of CRNs.

In this thesis, we explore potential threats to CRNs and provide a set of countermeasures to detect and mitigate them. In order to have a comprehensive understanding of their security concerns, next section provides an insight of the basis of CR technology and CRNs. Section 1.2 introduces the challenges that must be faced in order to secure CRNs and outlines the main goals of this thesis. Finally, Section 1.3 describes the structure of this document and the contributions of this work.

1.1 An overview of Cognitive Radio Networks

1.1.1 Cognitive Radio

A CR, a term coined by Mitola in [Mitola, 2000], can be defined as *an smart radio device which senses the RF environment, makes intelligent decisions based on sensing measurements and stored past data, and reconfigures itself accordingly.*

This definition hints at the two main nowadays accepted characteristics of CRs [Akyildiz et al., 2006]: 1) cognitive capability, which refers to the ability of the CR to capture information from its radio environment and identify the best spectrum portion and operating parameters such as modulation, coding, power transmission, etc; and 2) reconfigurability, that enables the radio components to be dynamically programmed according to cognitive decisions.

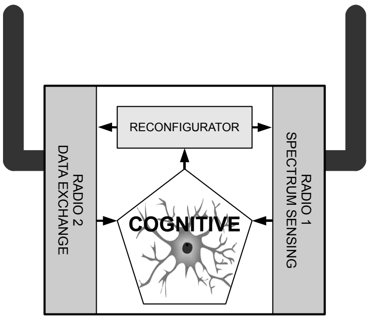


Figure 1.2: A cognitive radio device

Both characteristics are clearly shown in Figure 1.2 which conceptually represents a CR device with its basic modules. The cognitive module requires the use of sophisticated techniques (e.g. artificial intelligence algorithms [Zhang and Xie, 2007, Clancy et al., 2007, He et al., 2010]) in order not only to monitor the existence of transmissions in a given frequency, but also to capture the spatial and temporal variations in the radio environment. The reconfigurator module adjusts radio parameters based on the cognitive module decisions. On the other hand, two radio antennas are required in order to send/receive data and perform spectrum sensing simultaneously. Although CR technology is still maturing, there are several testbed implementations [Tkachenko et al., 2006, Tech, 2011, Crohas, 2008] which prove the feasibility of these systems.

1.1.2 Cognitive Radio Networks

A Cognitive Radio Network can be defined as a *network of CRs where all cognitive functions are performed in order to improve the end-to-end performance of the whole system.* Upon the proposal of the FCC [FCC, 2003b], research in CRNs has mainly focused on the design of mechanisms in order to identify vacant portions of the spectrum or 'white spaces' and efficiently access to these spectrum without causing harmful interference to licensed users.

Consider the network model shown in figure 1.3; a *Primary Network* is an existing network infrastructure operating with license in a given spectrum band, such as current cellular or TV broadcast networks and offering its services to incumbents or *Primary Users.* In this context, a CRN is allowed to operate in both licensed and unlicensed bands. When using the licensed band, the CRN may coexist with existing primary networks, and thus it is seen as a *Secondary Network*, where secondary users are allowed to use the spectrum in an opportunistic manner. Therefore, a CRN must perform spectrum sensing in order to identify the portions of spectrum left unused by primary networks and, among those available, select the channel with best conditions. During its operation in a given channel, a CRN must keep on performing spectrum sensing and, whenever a primary transmission is detected in the current band of operation, it must immediately vacate the channel and switch to another one, a process known as spectrum handoff. Note that many CRNs may overlap trying to make use of the spectrum left by a primary network, also referred as *self-coexistence*. As a consequence, there is also a need for mechanisms to enable coexistence among overlapping CRNs.

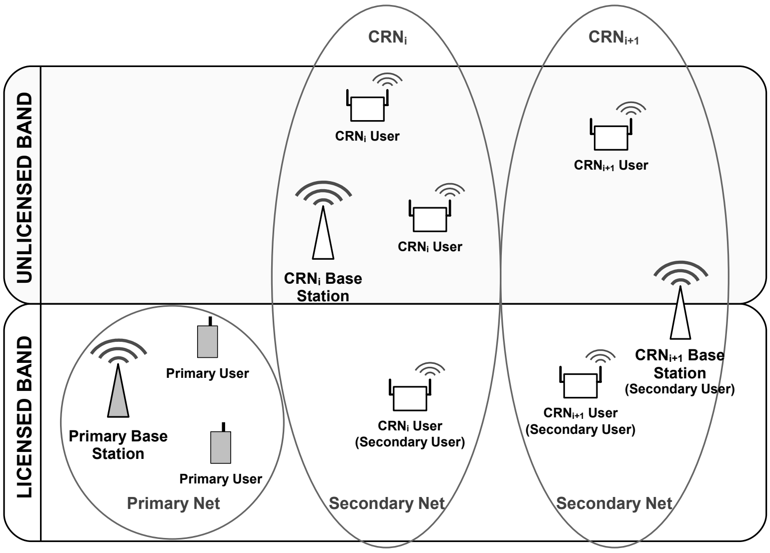


Figure 1.3: CRN model

To summarize, the new functionalities required in CRNs are as follows:

- Spectrum sensing: It is needed to sense the medium in order to detect unused spectrum and avoid causing harmful interference to legitimate users of the spectrum.
- Spectrum management: A CRN must select the best available channel to meet user communication requirements.
- Spectrum mobility: A channel must be vacated when a licensed user is detected while maintaining seamless communication requirements during the transition to better spectrum.
- Spectrum sharing: A fair spectrum scheduling method among coexisting secondary users must be provided.

These functions are closely related to each other and form the Cognitive Radio cycle shown in Figure 1.4. As an example, the spectrum sensing function is responsible for gathering information from the environment. These data is then used by the spectrum management function in order to decide how to allocate spectrum bands. On the other hand, such functions can be performed in a collaborative way, i.e., the CRs can exchange information among them, or they can be implemented by each CR on its own. Although the collaborative approach

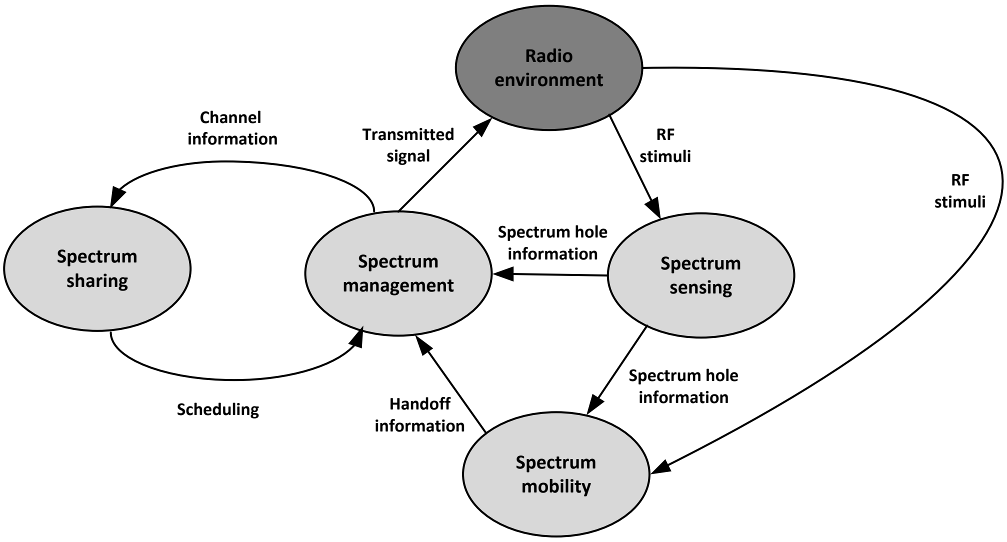


Figure 1.4: Cognitive Radio cycle

requires the use of a common channel to exchange information, it is considered more efficient [Baldini et al., 2011]. As a consequence, cooperation in CRNs is often assumed in the literature.

CRN architecures

Generally speaking, CRNs can be classified into two main categories according to their architecture:

- Centralized CRNs
- Distributed or non-centralized CRNs

In centralized networks there is an only top level entity in charge of the network management. An example of centralized network is an infrastructure-oriented network composed by a Base Station (BS) controlling a given cell, where decisions about spectrum access are taken by the former. Users within that cell perform spectrum sensing and periodically provide feedback to the BS which collects the data in order to take a global decision. On the other hand, in a distributed network the management is shared by several entities. When these entities are a subset of all the network entities then we talk about partially-distributed networks. When the management is shared out by all the network entities then we talk about fully-distributed networks. Fully-distributed networks are often based on Peer-to-peer (P2P) relations where users communicate in an ad-hoc manner and may exchange spectrum sensing information.

Regardless of their architecture, CRNs can also be classified according to the mechanism used to exchange sensing information: in band, if the data channel is also used to transmit control information, or out-of-band, if a dedicated channel is used with such purpose.

Several approaches have been proposed in the literature, such as the IEEE 802.22 [Cordeiro et al., 2006, 802.22 WG, 2011], DIMSUMnet [Buddhikot et al., 2005], KNOWS [Yuan et al., 2007], CORVUL [Mishra, 2004] or DSAP [Brik et al., 2005]. Table 1.1 shows a brief overview of the architectures previously referenced. For each one, we specify the type of network according to how spectrum allocation is performed, the type of coexistence mechanism, if used, and the type of channel used to exchange spectrum information.

Architecture	Spectrum Allocation	Self-coexistence	Control Channel
802.22	Centralized	Distributed (Coexistence Beacons)	In/Out-of-band
DIMSUMnet	Centralized	Centralized (Bandwidth Broker)	Out-of-band
KNOWS	Distributed	Not considered	Out-of-band
CORVUL	Distributed	Not considered	Out-of-band
DSAP	Centralized	Not considered	Out-of-band

Table 1.1: CRN architectures

Among them, it is worth mentioning the IEEE 802.22 proposal as an example following the centralized or non-distributed approach. In Section 1.1.4, we provide a more detailed description of this standard which was recently approved in July 2011.

On the other hand, note that the different functionalities of a CRN will strongly depend on its architecture. As an example, spectrum management and sharing may be controlled by the BS in centralized CRNs, or by a CR in coordination with its neighbors in fully distributed CRNs. With regard to spectrum sensing, decisions may be taken by the BS in centralized networks. In distributed networks, this task may be performed by a CR based on its own sensing information and on that received from neighboring nodes.

Although the four functionalities previously mentioned are indeed essential in any CRN, spectrum sensing is perhaps the most critical task since its disruption prevent the other functions from performing effectively. Because of this, next section is devoted to provide an overview of the existing spectrum sensing mechanisms.

1.1.3 Spectrum sensing

In a CRN, it is essential to properly identify white spaces in the spectrum and avoid interferences to incumbents. The successful detection of primary users strongly depends of particular type of detector used to identify primary transmissions and, based on the output of such detector, on the rule applied in order to take a decision about the existence or absence of primary transmissions. The following sections present an overview of both detection techniques and decision methods that may be used in CRNs.

Signal detection

There exist several mechanisms to detect primary signals which can be classified into three main categories, depending on the amount of knowledge about the signal required by the method [Zeng et al., 2010]. In the following, we provide a brief description of these methods pointing out the pros and cons of each one:

1. Methods requiring both signal and power information:

    - *Matched Filter Detection.* A matched filter is obtained by correlating the unknown signal with a known signal or a template. When there is prior knowledge of the primary signal, it is the optimal detection method.
    - *Radio identification based detection.* This category covers a set of techniques which are based on extracting several features from the signal such as frequency, transmission range, modulation technique, etc. One of these techniques is cyclostationary detection which exploits the periodic properties of the received signal that cannot be found in random noise or interferences. For example, Advanced Terrestrial System Committee (ATSC) signals can be identified because they are digital signals with a symbol rate of 10.76 MHz. Other techniques which can be applied specifically to detect ATSC signals are PN 511 sequence detectors and pilot detectors. The first is based on identifying a 511-symbol long PN sequence that is inserted in the data stream every 24.2 ms. The latter relies on the 8-VSB modulation used by these signals.

      In general, these techniques are more robust to noise uncertainties than others such as energy detection but are rather complex and require a significantly long observation time.

2. Methods requiring only noise power information (semi-blind detection):

    - *Energy Detection.* In this approach the received signal strength is measured and compared to a threshold to determine if the channel is idle or not. Its performance is poor under low Signal-to-Noise Ratios (SNRs) and cannot discriminate between signals, i.e. cannot distinguish between a primary signal and interference signals coming from other secondary users. However, it is considered the optimal detection method when there is no *a priori* knowledge of the primary signal.
    - *Wavelet-based sensing.* This technique employs the wavelet transform in order to detect portions of the spectrum being occupied and identify the singularities of the received signal. Its main drawback is that requires high sampling rates in order to characterize the bandwidth.

3. Methods requiring no information on source signal or noise power (blind detection):

- *Covariance-based sensing.* This method relies on the fact that statistical covariance matrices of the signal and noise are different, thus, allowing to effectively distinguish a primary signal when it is present.

The selection of the optimal method depends on several parameters such as the SNR, the type of signal to be recognized or the available amount of time for the detection process. Generally, the preferred technique is energy-detection because of its simplicity and its relatively good performance. Next section is devoted to describe how, based on energy measures, decisions are made regarding primary detection.

Decision methods

Given the output of an energy signal detector, hypothesis testing is usually applied in order to determine whether a primary user is present or not. The spectrum decision problem can be posed as a hypothesis test as follows [Digham et al., 2007]:

$$y_i = \begin{cases} n(t) & H_0 \quad \mathit{Idle} \\ hs(t) + n(t) & H_1 \quad \mathit{Occupied} \end{cases} \tag{1.1}$$

where y_i is the output detector, i.e., the signal received by the secondary user i, $s(t)$ is the transmitted signal, $n(t)$ is the white Gaussian noise and h is the amplitude gain of the channel.

In order to evaluate the performance of the detection mechanism, two metrics are generally used: the probability of detection $P_{d,i}$, which denotes the probability of CR i declaring that a primary user is present when the spectrum is occupied by a primary user, and the probability of false alarm $P_{fa,i}$, which denotes the probability of CR i declaring that a primary user is present when the spectrum is actually free. These probabilities can be expressed, respectively, as in (1.2) and (1.3),

$$P_{d,i} = P\left(decision = H_1|H_1\right) = P\left(y_i > \lambda|H_1\right) \tag{1.2}$$

$$P_{fa,i} = P\left(decision = H_1|H_0\right) = P\left(y_i > \lambda|H_0\right) \tag{1.3}$$

where y_i is the decision statistic, λ the decision threshold and H_0 and H_1 denote the hypothesis of absence or existence of a primary user in a given frequency.

Low values for $P_{d,i}$ result in missing the presence of a primary user with high probability and can lead to interferences to primary transmissions. On the other hand, a high $P_{fa,i}$ results in low spectrum utilization, since secondary users miss spectrum opportunities.

It can be shown that y_i follows a non-central χ^2 distribution under H_0 and non-centralized χ^2 distribution under H_1, with $2m = 2TW$ degrees of freedom, where TW is the time-bandwidth product and γ_i the reception SNR [Digham et al., 2007]:

$$y_i \;=\; \begin{cases} \chi^2_{2m} & H_0 \\ \chi^2_{2m}(2m\gamma_i) & H_1 \end{cases} \tag{1.4}$$

In a non-fading environment where h is deterministic, the detection and false alarm probabilities can then be computed as follows [Digham et al., 2007]:

$$P_{d,i} = P\{y_i > \lambda | H_1\} = Q_m\left(\sqrt{2m\gamma}, \sqrt{\lambda}\right) \tag{1.5}$$

$$P_{fa,i} = P\{y_i > \lambda | H_0\} = \frac{\Gamma(m, \frac{\lambda}{2})}{\Gamma(m)} \tag{1.6}$$

where Q_m is the generalized Markum function, and $\Gamma(.,.)$ and $\Gamma()$ are the incomplete and the complete gamma function respectively.

These probabilities strongly depend on the detected SNR and therefore a single CR may fail to detect the presence of a primary signal due to fading or shadowing. However, it is unlikely that all secondary users within a CRN experience fading or shadowing concurrently. Therefore, cooperation among CRs can help to improve the probability of detection by taking into account sensing reports from several CRs. Cooperation provides with space diversity and overcomes the limitations introduced by bad environment conditions in a particular area of the network.

In cooperative spectrum sensing, each secondary user performs its own measure and reports it to a fusion center, which combines the data and takes a global decision. The fusion center can be centralized or distributed; when it is centralized, all the CRs sent the reports to a single fusion center, e.g. the BS, which is responsible for taking a global decision and report back the results to the members of the CRN. If it distributed, every CR can act as a fusion center receiving sensing information from the neighboring nodes and taking a decision locally. Then, individual decisions can be shared among CRs and converge by iterations to a unified decision.

On the other hand, sensing reports may contain the observed data by a particular receiver, an approach known as data fusion (or soft-decision), or its individual decision about the existence of a primary, known as decision fusion (or hard-decision). Data fusion generally provides better results in terms of primary detection but requires the transmission of a high volume of data. Binary fusion reduces the amount of transmitted data because each node only needs to send one byte of decision: 1 if a primary exists or 0, otherwise. Despite it, in most cases has poorer performance in terms of false positives and false negatives [Visotsky et al., 2005]. Next, we describe the most commonly used methods with regard to both approaches.

Binary fusion (hard decision) The most known and basic binary fusion methods are the OR rule, the AND rule and the "K out of M" rule. Given the probabilities $P_{fa,i}$ and $P_{d,i}$ provided by a set of nodes $i = 1...M$ and computed as in (1.3), the final probability of detection P_d and the final probability of false alarm P_{fa} can be derived for each approach as follows.

- **OR rule**. The final decision is “exists a primary (1) ” if at least one of M receivers reported “1”. The probability of false detection and false alarm is then given by $P_d = \prod_{i=1}^{M}(1 - P_{d,i})$ and $P_{fa} = \prod_{i=1}^{M}(1 - P_{fa,i})$
- **AND rule**. The final decision is “exists a primary (1) ” if all the M receivers reported “1”. The probability of false detection and false alarm is then given by $P_d = \prod_{i=1}^{M} P_{d,i}$ and $P_{fa} = \prod_{i=1}^{M} P_{fa,i}$
- **K out of M rule**. The final decision is “exists a primary (1) ” if at least K out of the M receivers reported “1”. The probability of false detection and false alarm is then given by $P_d = \sum_{i=0}^{M-k} \binom{M}{K+i}(1 - P_{d,i})^{M-K-i}$ and $P_{fa} = \sum_{i=0}^{M-k} \binom{M}{K+i}(1 - P_{fa,i})^{M-K-i}$

The OR rule works best, i.e., maximizes the detection probability and minimizes the false alarm probability, for a large number of cooperative users while the AND rule presents its best performance for a small number of users. The majority rule can be obtained by means of the k out of N rule, under the condition $k \geq N/2$. In such case, it is important to obtain the value of k for which the detection errors are minimized. It can be shown that this value depends on the detection threshold λ [Akyildiz et al., 2011].

Data fusion (soft decision) When data fusion is used, diversity techniques [Brennan, 2003] are typically applied in order to combine the data and take a decision. The simplest one is Equal-Gain Combining (EGC) [Digham et al., 2003], in which the fusion center compares the sum of N individual observations y_c with a global threshold λ_c as in (1.7).

$$y_c \quad = \textstyle\sum_{i=1}^{N} y_i \qquad \begin{matrix} \leq \lambda_c & H_0 \\ \geq \lambda_c & H_1 \end{matrix} \tag{1.7}$$

The probability of detection with EGC can be computed analogously to (1.6) as in (1.8), considering that the sum of N independent and identical distributed (i.i.d.) non-centralized χ^2 variables with $2m$ degrees of freedom results in another non-centralized χ^2 variable with $2Nm$ degrees of freedom.

$$P_{d,EGC} = P\{y_c > \lambda | H_1\} = Q_m\left(\sqrt{2Nm\gamma_t}, \sqrt{\lambda}\right) \tag{1.8}$$

where $\gamma_t = \sum_{i=1}^{N} \gamma_i$ is the sum of the SNRs for each receiver i.

A variant of this method is based on assigning a given weight to the received observation y_i from the ith user as in (1.9).

$$y_c = \sum_{i=1}^{N} w_i \cdot y_i \qquad \begin{matrix} \leq \lambda_c & H_0 \\ \geq \lambda_c & H_1 \end{matrix} \tag{1.9}$$

where $\mathbf{w}$ is a vector whose elements satisfy $\sum_{i=1}^{N} w_i = 1$, and every weight w_i can be assigned according to how reliable is a given measure.

There exist a variety of other methods which achieve better performance at the expenses of increasing in complexity and requiring additional information, that may be not available. As an example, Maximal-Ratio Combining (MRC) [Herath et al., 2011] requires the estimation of the fading-channel gain for each pair emitter-receiver.

1.1.4 The IEEE 802.22 WRAN standard

As a result of the release of the NPRM by the FCC [FCC, 2003b], the IEEE 802.22 Working Group (WG) [802.22 WG, 2011] was created in 2003 with the aim of defining the first wireless air interface standard based on CRs. This standard, which was recently approved, was mainly intended to provide wireless broadband access in rural and remote areas. Since the creation of the 802.22 WG, most of the research on CRNs has been done according to its guidelines. For this reason, in the following we give a brief overview of this standard.

The 802.22 WRAN system

The IEEE 802.22 Wireless Regional Area Networks (WRAN) standard [Cordeiro et al., 2006, 802.22 WG, 2011] defines a point-multipoint air-interface, composed by a BS and several CRs, referred to as Consumer Premise Equipments (CPEs). The WRAN system inherits many features from WiMAX but differs in some aspects, such as the service coverage and the fact that WRAN users must coexist with primary users which have priority to use the bandwidth. In 802.22, the BS coverage range can go up to 100 Km, although the current specified coverage range is 33 Km, corresponding to a transmission power of 4 Watts (W) CPE Equivalent Isotropically Radiated Power (EIRP). A WRAN may operate at frequencies in the range 54-862 MHz and the channels to be used have a bandwidth of 6,7 or 8 MHz, corresponding to different TV channel bandwidth, which depend on the country. For 6MHz, it represents a potential amount of 47 TV channels to be used by the WRAN. However, with the purpose of avoiding interferences to primary users, WRAN systems are neither allowed to operate at those channels in use by primary networks or at the adjacent channels, a fact that considerably reduces the number of available channels.

The standard also specifies the PHY and MAC layers requirements that must accomplish a given CPE. With regard to the PHY layer, 802.22 devices must use Orthogonal Frequency-Division Multiple Access (OFDMA) which provides several benefits over other access technologies, such as higher spectral efficiency or robustness to Inter-Symbol Interference (ISI). Besides, it is required flexibility in terms of coding and modulation (QPSK, 16-QAM and 64-QAM) to adapt to varying channel conditions, and effective power control to minimize interferences to primary users.

At the MAC layer, the 802.22 standard provides a connection-oriented service supporting

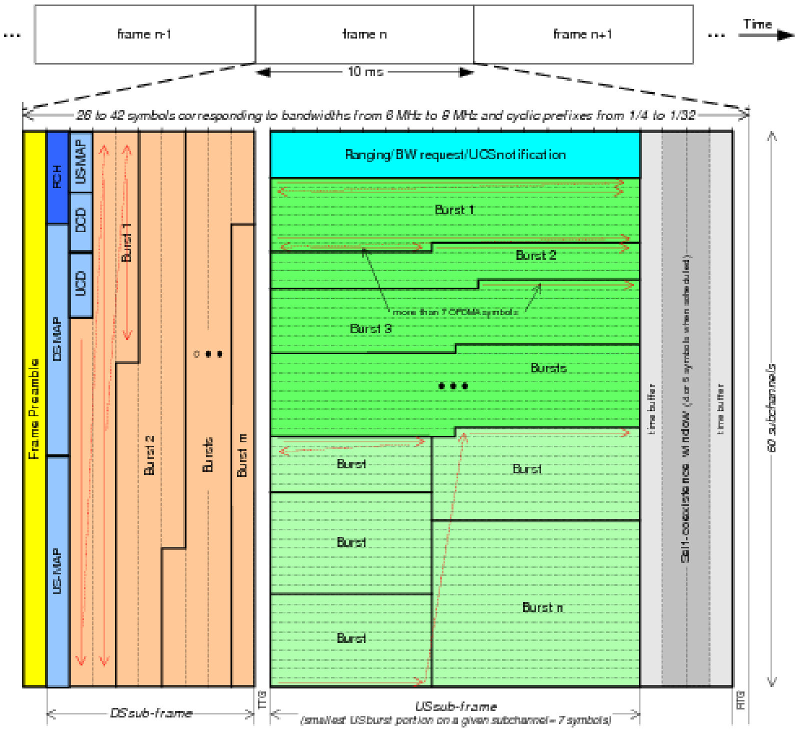

...
frame n-1
frame n
frame n+1
...
Time
10 ms
26 to 42 symbols corresponding to bandwidths from 6 MHz to 8 MHz and cyclic prefixes from 1/4 to 1/32
Frame Preamble
FCH
US-MAP
DCD
UCD
DS-MAP
US-MAP
Burst 1
Burst 2
Bursts
Burst m
Ranging/ BW request/UCS notification
Burst 1
Burst 2
more than 7 OFDMA symbols
Burst 3
Bursts
Burst
Burst
Burst
Burst
Burst n
time buffer
Self-coexistence window (4 or 5 symbols when scheduled)
time buffer
60 subchannels
DS sub-frame
TTG
US sub-frame
(smallest US burst portion on a given subchannel = 7 symbols)
RTG

Figure 1.5: IEEE 802.22 frame structure [2]

several types of QoS. Communication between CPEs and the BS is performed using a Time Division Duplex (TDD) structure: a superframe of 160ms consisting in 16 frames of 10ms. At the beginning of each superframe, the BS sends a special preamble through each and every TV channel that can be used for communication, i.e., that meets incumbent protection requirements. CPEs tuned to any of these channels are then able to obtain the necessary information to associate with the BS.

The frame structure is depicted in Figure 1.5. In the time domain, each frame is composed of a number of symbols ranging from 26 to 42, depending on the TV channel bandwidth (6, 7 or 8 MHz) and the Cyclic Prefix (CP) (from 1/4 to 1/32), which is used to minimize the effect of the ISI. In the frequency domain, each frame is composed of 60 subchannels, where two of them are reserved for special purposes, such as bandwidth demand. The minimum unit is an OFDMA slot, corresponding to a single OFDMA symbol by one sub-channel. As it can be seen, the frame is also divided into a Downlink (DL) subframe and an Uplink (UL) subframe. The DL subframe is used to send data from the BS to the CPEs, and contains the necessary information for CPEs to synchronize (frame preamble) and the bandwidth allocation established by the BS. The UL subframe is used to transmit data from the CPEs to the BS, according to the bandwidth allocation established in the DL subframe. Slots within the UL subframe are grouped into bursts, which define a logical subchannel composed by one or several subcarriers, a set of symbols and a given modulation. A single burst may contain

[2] Image taken from [802.22 WG, 2011]

several Medium Access (MAC) Protocol Data Units (PDUs) from different CPEs.

Primary protection

There are two types of primary users defined in the standard: TV towers and wireless microphones. A WRAN must vacate a channel if licensed signals are detected above the following thresholds: -116dBm over a 6MHz channel for digital TV and -107dBm for wireless microphones measured in a 200KHz bandwidth. The value of these thresholds implies that CPEs should be able to detect primary transmissions at very low SNRs. On the other hand, with the aim of protecting primary transmissions, three mechanisms are defined in the standard:

- Sensing: the BS shall instruct its CPEs to perform in-band or out-of-band measurements in order to detect primary transmissions. For in-band measurements, the BS may establish quiet periods in which no CPE is allowed to transmit, with the purpose of effectively detecting the presence of incumbents. Quiet periods can be established either between consecutive superframes or consecutive frames. On the other hand, two different types of sensing have been defined: fast sensing and fine sensing. During the fast sensing stage, a sensing mechanism such as energy-level may be used to detect primary signals rapidly (around 1ms per sensed channel). Based on the set of values provided by the set of CPEs, the BS may ask them to perform fine sensing in order to obtain detailed features of the detected signal, for example by means of a cyclostationary detection technique. Generally, it require a longer sensing time than fast sensing techniques.

- Database access: The BS may store information about the location of primary transmitters, e.g. TV towers, and about CPEs. In fact, the standard suggests that each CPE should provide geolocation information and device identification when associates to the BS. Note that the position of the CPEs can be useful to locate a given transmission and verify its legitimacy .

- Beacons: Given that the position of wireless microphones is not fixed, as it is the case of TV towers, the solution adopted for protecting wireless microphones is based on beacon signals. These signals should be transmitted from the wireless microphones to the BS with a power 250mW, a considerably higher value than the power of 10mW used for normal transmission.

The standard defines a minimum probability of detection and a maximum probability of miss detection of primary users of 0.9 and 0.1 respectively. Besides, when a primary is detected, the WRAN must perform a spectrum handoff following the Dynamic Frequency Selection (DFS) timing parameters ordered by the FCC [FCC, 2003a]. The key parameters are the Channel Detection Time (CDT), the Channel Move Time (CMT) and the Channel Closing Transmission Time (CCTT). The CDT is the time needed in order to effectively recognize a primary signal and must be below 2s. The CMT refers to the amount of time elapsed since the primary is detected until the WRAN completes the frequency handoff and is set to 2s. Finally, the CCTT represents the amount of transmission time during the CMT and is set to 100ms.

Self-coexistence

As multiple 802.22 networks may coexist in the same area, there is a need for mechanisms to avoid interferences among them. With this purpose, the IEEE 802.22 standard defines a Coexistence Beacon Protocol (CBP). Coexistence packets are transmitted at the end of some frames by the BS and/or some designated CPEs and monitored by other BSs and CPEs operating on the same or different channels. These packets contain information about the cell and the candidate channels to be used by a given WRAN, and thus can be used to avoid overlapping of several WRANs in the same channel. Moreover, they can also be used to synchronize different WRANs for quiet periods with sensing purposes.

Security Sub-layer

The IEEE 802.22 standard defines a MAC security sublayer which provides basic security services by applying cryptographic transformations to data units exchanged between the BS and the CPEs. This security sublayer has two main components: an encapsulation protocol and a Privacy Key Management (PKM) protocol. The encapsulation protocol defines a set of supported cryptographic suites and the rules for applying these algorithms to a MAC PDU (MPDU). Confidentiality and integrity is achieved by means of AES-GCM and CPE network entry authorization is obtained through RSA and ECC based X.509 certificates. On the other hand, the PKM protocol ensures the secure distribution of keying material from the BS to the CPEs. With regard to signals such as beacons and CBP, ECC based digital signatures are used to authenticate these packets but no encryption is provided, thus, making information about the WRAN operation accessible to malicious users. Finally, the standard also recommends Trusted Platform Module (TPC) specifications to enable tamper-proof capability for hardware and software of CPEs.

1.2 Security in CRNs

As any other type of wireless network, CRNs make use of an open communications medium which can be easily accessed by malicious users. Therefore, security becomes an indispensable element to ensure the desired level of network performance. As a general rule, CRNs should guarantee basic security services such as data confidentiality and privacy, integrity, availability, authentication and authorization by means of cryptographic primitives. However, CR technology offers new capabilities leading to new security holes that cannot be properly addressed by such countermeasures.

Security attacks could be implemented against the CRN functionalities by providing wrong information on the radio environment or by disrupting the cognitive mechanisms. For example, an attacker altering the medium could lead to erroneous decisions regarding the availability or the quality of a given channel, and therefore to an incorrect management of the spectrum. On the other hand, the learning algorithms used by the cognitive engines embedded in CRs could be disordered either by feeding them with false data, or by illegally manipulating of the software.

Special consideration should be also devoted to those mechanisms used in cooperative CRNs. Generally, such approaches are built on the basis that nodes are honest and altruistic, making CRNs vulnerable to security threats such as selfish or malicious behaviors. A selfish node which do not cooperate to its own benefit, or compromised nodes which report false data could degrade the performance of the network with regard to spectrum utilization. Besides, an attacker could completely disrupt the CRN operation by preventing CRs from communicating.

Being aware of the vulnerable nature of CRNs, the purpose of this work is to contribute to the enhancement of the security of such networks. In particular, the main goals of this thesis can be summarized as follows:

- Fully understand the mechanisms involved in a CRN operation and its security vulnerabilities.
- Identify new threats to CRNs arising due to their particular features.
- Evaluate the impact of the main attacks on CRNs performance.
- Design efficient mechanisms to detect and counteract these attacks.

1.3 Structure of the thesis

This thesis is composed of eight chapters structured in four parts. The first part includes one single chapter which corresponds to this introduction, and provides an overview of CR technology and the main features of a CRN, being the scenario in which this thesis has been developed. In Section 1.2, we present the main objectives of this work. Finally, the present section outlines the remaining of the document and the different contributions.

Following the introduction, Part II provides a review of the State of the Art regarding security in CRNs and is divided into two different chapters. Chapter 2 is devoted to describe the main threats to CRNs identified in the literature while in Chapter 3, we give an insight of the different mechanisms proposed to counteract such attacks.

Part III presents the contributions of this work and is composed of four chapters. Chapter 4 provides a set of guidelines in order to effectively achieve a high level of security in CRNs, which have been followed along the development of this thesis.

In Chapter 5, we propose a new cross-layer attack to the Transmission Control Protocol (TCP) connections established in a CRN. In order to have a comprehensive understanding of such attack, we first provide an insight of the basis of the TCP protocol. Next, the attack description and an analytical model of the former are provided. The degradation of TCP performance under this attack has been evaluated analytically and by means of simulation. Finally, we propose a set of mechanisms to counteract the attack. One of them is a modification to the basic TCP protocol, whose efficiency in mitigating the attack is proved via simulation.

Chapter 6 proposes to deal with one of the most known attacks to CRNs, i.e., the Primary User Emulation (PUE) attack, by means of a cooperative localization method. First, a general description of the main location techniques is given and their suitability for CRNs is discussed. Next, a location method tailored to the particular attributes of CRNs is provided. In this method, the CRs belonging to the network report to the BS time measurements in order to estimate the position of a given emitter. Based on this estimation, it is possible to discriminate between a PUE attack and a legitimate transmission. The effectiveness of the proposal is evaluated by means of simulation.

The cooperative location method presented in Chapter 6 can be disrupted by malicious CRs reporting false measurements, i.e., liar nodes. Several approaches have been proposed in the literature in order to counteract the effect of false data on cooperative methods, which are reviewed at the beginning of Chapter 7. Next, a scheme based on outliers detection techniques is proposed in order to detect abnormal reports, and make the location process robust to a given number of liars. The method is evaluated and compared to the basic location method via simulation, with regard to location accuracy and computational cost.

Finally, Part IV is composed of one single chapter, i.e., Chapter 8, which is devoted to summarizing all the achievements of this thesis, and also to present possible ways of extending this work in the near future.

1.3.1 Contributions

To sum up, the original contributions result from our work during the development of this thesis, are identified in the following five points:

- A critical analysis of the State of the Art regarding security in CRNs. The results of this study were published in [León et al., 2010].
- A roadmap outlining the steps to follow in order to effectively secure CRNs. This work was published in [León et al., 2011b].
- Proposal of a new attack to TCP in CRNs. An analytical model has been derived for the attack. Its impact on the performance of TCP connections has been evaluated both by means of simulations and the analytical model. Several countermeasures have been provided in order to detect the attack and mitigate its effect. The work presented in [León

et al., 2009a, León et al., 2009b] provides a description of the attack and its evaluation by means of the simulator ns-2 [USC/ISI et al., 2007]. Besides, the improvement introduced by one of the proposed countermeasures is also analyzed via simulation. The analytical model was presented in [Hernández-Serrano et al., 2010] and validated by comparing its results with those obtained from the simulations.

- A localization method as countermeasure for PUE attacks. This method is based on the Time Difference of Arrival (TDoA) technique and relies on measurements performed by the members of the CRN in order to locate a given emitter. Based on the position estimation, a decision about the existence of an attacker or a legitimate primary transmitter is performed. The idea was first presented in [León et al., 2011b]. Full details of the mechanism and its evaluation have been submitted to [León et al., 2011].

- A false feedback detection scheme to be applied into the localization method. This method improves the robustness of the location process against a variable number of malicious nodes reporting false data. A draft of this scheme was first presented in [León et al., 2011b]. In [León et al., 2011a], a description of the mechanism and some preliminary results are given. A deeper evaluation in the context of WiMAX networks has been submitted to [Hernández-Serrano et al., 2011].

1.3.2 List of Publications

- [León et al., 2009a] León, O. and Hernández-Serrano, J. and Soriano, Miguel, ”Un nuevo ataque a TCP para redes de radios cognitivas”, VIII Jornadas de Ingeniería Telemática (JITEL’09), Cartagena (Spain), 09/2009, ISBN: 978-84-96997-27-1

- [León et al., 2009b] León, O. and Hernández-Serrano, J. and Soriano, Miguel, ”A new cross-layer attack to TCP in cognitive radio networks”, Second International Workshop on Cross Layer Design (IWCLD’09), 2009, pp.1-5, doi:10.1109/IWCLD.2009.5156526

- [León et al., 2010] León, O. and Hernández-Serrano, J. and Soriano, Miguel, ”Securing cognitive radio networks”, International Journal of Communication Systems, 2010, Vol. 23 n^o 5, pp. 633-652, doi:10.1002/dac.1102.

- [Hernández-Serrano et al., 2010] Hernández-Serrano, J. and León, O. and Soriano, M., ”Modeling the Lion Attack in Cognitive Radio Networks”, EURASIP Journal on Wireless Communications and Networking, vol. 2011, issn: 1687-1472, Hindawi Publishing Corporation.

- [León et al., 2011b] León, O. and Román, R. and Hernández-Serrano, J., ”Towards a cooperative intrusion detection system for cognitive radio networks”, NETWORKING 2011 Workshops, 2011, pp. 231–242, Springer.

- [León et al., 2011a] León, O. and Hernández-Serrano, J. and Soriano, Miguel, ”Robust Detection of Primary User Emulation Attacks in IEEE 802.22 Networks”, 4th International Conference on Cognitive Radio and Advanced Spectrum Management (CogArt 2011), 2011, ACM digital library.

- [León et al., 2011] León, Olga and Hernández-Serrano, Juan and Soriano, Miguel, ”Cooperative Detection of Primary User Emulation Attacks in Cognitive Radio Networks”, in Computer Networks (accepted with major revision)

- [Hernández-Serrano et al., 2011] Hernández-Serrano, J. and León, O. and Soriano, M., ”Robust localization in WiMAX in the presence of malicious anchor nodes”, submitted to Computers and Electrical Engineering.

Part II

State Of The Art

Chapter 2

Threats to CRNs

Wireless networks are growing in popularity due to its easy deployment and their ability to provide high-speed access to portable devices, and to areas where running cable is not an option. However, malicious users can also take advantage of such ease of access and mobility to attack from any location. As wireless communications use the air as the physical media, they are more easily accessed by an attacker and thus intrinsically more vulnerable than its wired counterparts to attacks such as eavesdropping, data modification, impersonation or Denial of Service (DoS).

Due to its wireless nature, CRNs inherit most of the threats already reported in the literature in the context of wireless networks. However, the flexibility and reconfigurability capabilities of these networks can make them even more sensitive to conventional attacks but also expose them to new security implications [Clancy and Goergen, 2008, Burbank, 2008]. Generally speaking, security in CRNs has received less attention than other areas of CR technology such as spectrum sensing or spectrum management. Because of this, there is a clear need for identifying potential new threats and analyzing their impact on the performance of CRNs.

In the context of CRNs, we define an attack as an action that achieves at least one of the following goals:

- **Unacceptable interference to licensed primary users**. Because of the attack, the communication channel of the primary/licensed users is diminished or becomes unusable, i.e., a DoS attack.

- **Missed opportunities for secondary users**. An attacker could prevent secondary users from using available spectrum bands, by reducing the channel performance or denying service to secondary users, among others.

- **Access to private data**. An attacker could try to access data without authorization. As a consequence data must be secured by cryptographic primitives.

- **Modification of data**. An attacker could try to modify the data exchanged between several entities to its own advantage. Thus, integrity of data must be assured.

- **Injection of false data**. Injection of false data could lead the CRN to behave in an unpredictable way or to follow the attacker guidelines. Therefore, authentication of information sources must be guaranteed.

As in any other type of network, the last three threats may be overcome by providing basic security services such as confidentiality, integrity and authentication. However, as these approaches have been broadly studied and spread across several scenarios without significant changes, we will mainly focus on those which are related to the CRNs nature. These specifics attacks are classified into inside and outside depending on whether the attacker is a member of the CRN (insider) or not (outsider). Furthermore, their impact on the different CRN architectures is thoroughly discussed.

2.1 Outside attacks

Outside attacks are those carried by an entity from outside the victim CRN, i.e., attacks that can be executed by a non-authorized entity. As a result, we assume the attacker cannot log in (spoof) the victim network. Obviously, when the attacker spoofs an authorized identity can also execute insider attacks (those performed by authorized entities).

Notice that it cannot be assumed that an outside attacker, although it is not authorized, has no knowledge about the victim network. Quite the opposite, the attacker may know the sensing protocols, the potential primary users, etc. and hence execute more specific attacks.

2.1.1 Jamming

Within the simplest form, a jamming attack involves the radiation of radio signals that intentionally[1] disrupt communications in the victim network. If the generated interferences are big enough, they can substantially decrease the performance of communications or completely interrupt them, thus implementing a DoS attack. A study of the impact of jammers on CRNs is conducted in [Li and Cadeau, 2011]. Due to the characteristics of CRNs, a single jammer can reduce the throughput to 50%-70%, meaning that a few jammers attacking the network simultaneously could easily lead to a DoS . Despite it, a DoS attack for CRNs must cover all the potential frequencies and hence it is harder to perform than for other wireless networks with fixed frequencies.

In CRNs, interferences can be also created by “pseudo” primary users and as a consequence, detecting the attack may be harder than in conventional wireless networks. In the following, we present a set of specific jamming attacks to CRNs.

[1] The term *jamming* is used for intentionally disruption of communications while the term *interferences* usually refers to unintentionally one

Primary User Emulation attacks

As mentioned in Section 1.1.2, CRNs act as secondary users of the licensed spectrum and must not interfere with primary transmissions. An adversary could take advantage of this feature in order to disrupt communications within a CRN by performing a PUE attack. In a PUE attack, first coined in [Chen and Park, 2006], an attacker pretends to be a primary user or incumbent by transmitting a signal with similar characteristics to a primary signal or replying a real one. If the attack succeeds, it prevents the CRN from using a vacant band.

The impact of the PUE attack depends on several factors, such as the location of the attacker or the sensing mechanism used by the CRN. Selecting an optimal position to perform the attack will cause many secondary users reporting the existence of a primary transmission, and therefore will lead the CRN to look for another portion of the spectrum. On the other hand, if the sensing mechanism used by the CRN looks for specific characteristics of the signal, as it is the case of PN5 sequence, pilot detector or cyclostationary detection (see Section 1.1.3), the fake signal should fulfill several requirements with regard frequency, code, modulation, etc., in order to appear as a legitimate one. Although in this case the PUE attack is harder to perform, it is still quite feasible because the attacker can program its CR device in order to match the transmission parameters of a primary user, or even transmit a real primary signal previously recorded.

The most widespread sensing mechanism used for primary detection, however, is energy detection [Cabric et al., 2004], which simply relies on an energy threshold in order to take a decision about the existence of a primary. Because energy detection is unable to discriminate between primary and secondary signals, the 802.22 standard suggests the use of quiet periods in the CRN in which transmissions are not allowed. This allows performing spectrum sensing without the potential interferences that secondary users could produced. Thus, any transmission detected during that period may be considered as a primary signal if the received power at CRs is above a given threshold. This threshold plays an essential role since the lower the threshold, the higher the detection probability but also the easier to perform a PUE attack.

Moreover, with previous knowledge on the CRN operation, the attacker can force PUE attacks whenever the CRN switches from one channel to another, leading to a DoS. The attacker can gather information about the channel being used by the CRN in the following ways:

- By performing spectrum sensing until finding the new channel of operation of the CRN. In order to minimize the search time, the attacker may discard some channels directly, e.g., channels already in use by primary users. Moreover, if the attacker lies in the CRN area, it can estimate the most likely CRN channel by means of its own sensing measures.
- By eavesdropping the common control data of the CRN (if exists). This threat can be easily overcome by securing the channel by means of cryptographic primitives, as it is

already recommended by the IEEE 802.22 standard.

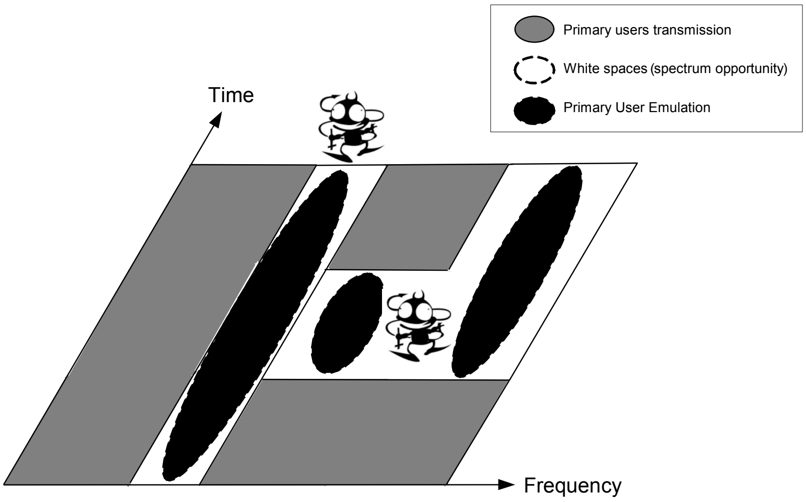

Primary users transmission
White spaces (spectrum opportunity)
Primary User Emulation
Time
Frequency

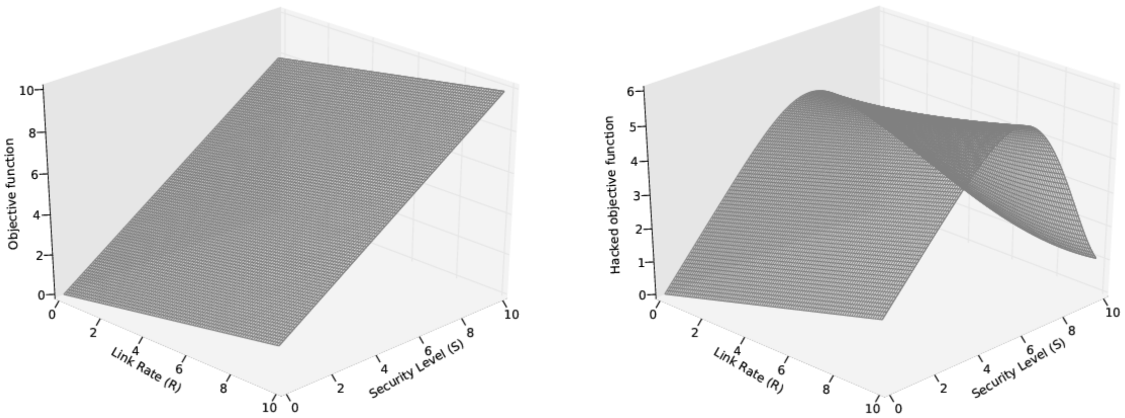
Figure 2.1: PUE attack

In [Anand et al., 2008], an analytical model which studies the feasibility of a PUE attack is presented. The authors derive mathematical expressions for the probability of a successful PUE attack and provide lower bounds on the probability of a successful attack using Fenton's approximation and the Markov's inequality. The simulation results show that the probability of a successful attack increases with the distance between real primary transmitters and secondary users belonging to a given CRN.

The worthiness of performing a PUE attack against a simple jamming attack is studied in [Peng et al., 2009] and [Peng et al., 2011]. Peng et al. present an attack model where an adversary can minimize the throughput of secondary users by combining PUE and jamming attacks, while spending a constraint amount of resources, i.e., energy. In this proposal, the attacker is assumed to have knowledge on the CRN operation. In this way, it intelligently performs PUE attacks during sensing periods, in order to decrease the number of available bands for the CRN, and performs jamming attacks during transmission periods, so that the throughput of secondary users is minimized.

Attacks to the learning engine

One of the key features of CRs is their capability of adapting to changing conditions in order to enhance network performance, based on current observations of the environment and past experiences. Learning from the past requires the introduction of a learning engine capable of trying out different radio configurations and observe how the system performs. Usually, this is done by means of Artificial Intelligence (AI) algorithms, such as genetic, hill-climbing or random walks [Russell and Norvig, 2002, Zhang and Xie, 2007, Clancy et al., 2007, He et al., 2010]. These algorithms make slight modifications of several input factors to find the



(a) Objective function

(b) Hacked objective function after an OF attack

Figure 2.2: Objective functions

optimal values that maximize an objective or goal function. In the context of CRs, input factors can be frequency, bandwidth, power, modulation type, coding rate, channel access protocol, encryption type, authentication type, message integrity code and frame size [Clancy and Goergen, 2008].

When CRs are in their learning phase, an attacker can easily disrupt this process by degrading the channel. Within one of the simplest approaches, we can define an objective function with two goals: high-rate and security. Typically, the objective function is defined as a weight function in which the different requirements of each service will determine the weight assigned to each goal. For example, in multimedia transmissions high-rate would be assigned the highest weight; on the other hand, data transactions would give more priority to security parameters. With such goals, we could define an objective function as in (2.1), with R the link rate, S the predefined level of security and ω_i the weight assigned to parameter i.

$$f(R,S) = \omega_R R + \omega_S S \tag{2.1}$$

As an example, in secure data transactions the weights could be set to $\omega_R = 0.2$ and $\omega_S = 0.8$. Then, the CR would vary its radio parameters in order to find the maximum of the objective function. Figure 2.2a represents $f(R,S), R \in [0,10], S \in [0,10], \omega_R = 0.2, \omega_S = 0.8$.

Note that the security level S is defined as a user/policy input but on the contrary, the link rate R is more related to the channel conditions. Thus, by altering the channel an attacker could manipulate the link rate. By way of illustration, let us consider an attacker which is able to hack a given security level s_1 but not greater than it. In order to disallow higher levels of security $S > s_1$, the attacker could jam the channel whenever the security level is $S \geq s_1$. Then, the target of the adversary would be to alter the objective function as in (2.2).

$$f(R,S) = \omega_R R + \omega_S S < \omega_R r_1 + \omega_S s_1 \quad \forall S > s_1 \tag{2.2}$$

In other words, the attacker should jam the channel in order to maintain a transmission rate as in (2.3).

$$0 \leq R < r_1 + \frac{\omega_S}{\omega_R}(s_1 - S) \quad \forall S > s_1 \tag{2.3}$$

Let's suppose that because of the attack, the representation of the objective function is the one shown in Figure 2.2b. The CR would adjust its radio parameters until finding the values that maximize the misled objective function. However, far from getting the best results, the CR device would set the security level to s_1, which could be easily hacked by the attacker.

Note that learning algorithms can be fed with training data *a priori* (supervised or off-line learning) or by means of data acquired over time (unsupervised or on-line learning). Given the dynamics of wireless environments which frequently change due to phenomena such as noise, multipath, etc, unsupervised learning may provide better results at long term . However, they can also be easily manipulated by malicious users.

Learning algorithms are also used with other purposes, such as classification of signals for primary identification [Clancy and Khawar, 2009, Clancy et al., 2011]. Analogously, these mechanisms can be disrupted through the transmission of crafted signals, leading to a wrong perception of how real primary signals are and thus increasing the probability of success of subsequent PUE attacks.

2.1.2 Common control channel attacks

In some approaches, a dedicated channel is used to exchange sensing information: a) between the BS and the secondary users if the CRN is centralized (i.e. DIMSUMnet); b) among secondary users if it is distributed (such KNOWS or CORVUS). A malicious user could jam this channel, thus disrupting all transmissions and preventing the CRs from sharing information about spectrum usage. The lack of knowledge about available bands would keep the CRN from operating, i.e., a DoS attack. The impact of this attack may be higher in centralized CRNs, since an attacker can focus on jamming the control channel within the BS vicinity (single point of failure), thus affecting the whole network.

On the other hand, common control channels carry crucial data for the operation of the CRN such as frequency, coding and modulation to be used for communication, or sensing data. Therefore, these channels must be secured in order to guarantee confidentiality, authentication and integrity of the data by using cryptographic primitives. In this way, in order to avoid that an eavesdropper obtains information about the operation of the CR and takes advantage of it to perform an attack, such as PUE attacks.

2.1.3 Spoofing

When no secure authentication/authorization method is provided, an outside attacker can become an insider by spoofing a valid identity and perform any of the inside attacks described in

Section 2.2. For this reason, authentication/authorization at the link layer should be mandatory to secure the network.

The Sybil attack is a form of spoofing attack in which the attacker spoofs multiple identities[3], and hence it can gain a larger influence. This attack, which was originally intended to attack the redundancy mechanisms of peer-to-peer storing systems, is also commonly used to attack routing protocols, data aggregation mechanisms, voting systems, avoid detection of malicious users, etc.

2.1.4 Cross-layer attacks

Cross-layer attacks are those which exploit the vulnerability of a given layer but are targeted to disrupt an operation at a different layer. For this reason, these attacks are especially difficult to detect. However, the amount of cross-layer attacks that can be performed by outsiders is somehow limited, since they can only access the physical or the link layer.

Key depletion attacks

Many transport security protocols, such as Secure Sockets Layer (SSL) or Transport Layer Security (TLS), use stream ciphers instead of block ciphers because of their low-cost. Since a stream cipher key stream can never be reused, the session key is established at the beginning of each transport layer session.

Mathur and Subbalakshmi in [Mathur and Subbalakshmi, 2007] pointed out that too many sessions would increase "too much" the risk of using the same key twice, and thus facilitate the cryptanalysis in the same manner as it occurs in the case of Wired Equivalent Privacy (WEP). However, this assessment is, to the best of our knowledge, completely wrong. Walker itself, in his original technical report [Walker, 2000], clarifies why the same cryptanalysis does not apply to other security protocols based on stream ciphers. In WEP, e.g., a 128 bits key is formed by a shared secret of 104 bits and a public random initialization vector (IV) of 24 bits that changes in every session. However, in SSL, an entire random, e.g., 128-bit key is selected for every session. Consequently, in order to gain the same level of advantage, the attacker should collect samples from about $2^{\frac{128}{2}=64}$ SSL ciphertext streams compared to the $2^{\frac{24}{2}=12}$ needed in the case of WEP. This implies that the cryptanalysis of SSL is 2^{52} (more than 4,000,000,000,000,000) times more costly than the one for WEP. Moreover, the cryptanalysis of WEP is enhanced by the bad-use of the Rivest Cipher 4 (RC4) stream cipher and by the active injection of traffic because of the lack of timestamps, i.e., replay attacks. WEP is just the example of what to never ever repeat and current and incoming wireless security implementations have learned such a lesson.

[3] The name of the attack is related to a book entitled *Sybil* that studies the case of a woman with dissociative identity disorder.

2.2 Inside attacks

An inside attacker can perform any of the outside attacks presented in Section 2.1 but also others that take advantage of its authorization to use the network. In the following, we present an overview of such attacks and classify them according to the TCP/IP model layer in which they are performed.

2.2.1 Link layer attacks

Attacks to CRNs at the link layer are mainly targeted to the cooperative sensing mechanisms. As explained in Section 1.1.3, cooperation among multiple CRs can significantly improve the performance of spectrum sensing, assuming that secondary users are honest and willing to cooperate. However, a given node may report false measures or decide to not cooperate, sinking in selfish behavior.

Selfish behavior

In a cooperative framework, selfish behavior refers to a set of actions performed by a node in order to increase its benefit while decreasing the average benefit for the rest of participants. Many studies have dealt with such problem targeted to ad-hoc routing [Roughgarden and Tardos, 2002, Qiu et al., 2006] but it has received less attention in the context of CRNs. In these networks, sharing of sensing measures among CRs is known to improve the performance of spectrum sensing. Therefore, selfish behaviors may increase the probability of wrong decisions regarding spectrum opportunities. As sensing results are usually broadcasted, selfish CRs may not cooperate in the sensing process in order to save energy while taken profit of sensing results provided by well-behaved neighbors.

The impact of this attack will obviously depend on the number of cooperating nodes and the correlation of their sensing measures. As an example, let us consider a CRN in which a single selfish node lies in a relatively big area with no neighbors, and is the only node which can detect the presence of a primary transmission. If it does not report the presence of the incumbent, it can lead to a false negative and cause interferences to primary users. Note, however, that this behavior can be easily detected in centralized CRNs where the BS controls the feedback provided by the set of CRs.

False Feedback (FF) attacks

Within a cooperative CRN, false feedback from one or a group of malicious users can lead the CRN to take improper actions and violate the goals of the protocol. As an example, a malicious secondary user could mislead spectrum sensing measures resulting in interferences to primary users, or inefficiency in spectrum usage, i.e., it can prevent the CRN from using a vacant band. This risk is especially relevant for fully distributed CRNs because the false feedback can be propagated affecting a large portion of the network. Such an effect is often referred as a *virus* due to its undesired distribution, but opposite to the "traditional" virus which refers to the application layer, it applies to the link layer.

In infrastructure-based networks such as IEEE 802.22, the BS usually acts as sensing data collector and applies a fusion method, either binary or data based, in order to take a decision. The impact of the attack will be minimal when there is a single malicious node and the total amount of cooperating nodes is large, but one must also consider the possibility of colluding nodes working together against the cooperative mechanism.

Besides, the effect of the attack will also depend on how sensing reports are forged: randomly or deterministically. Although modifying reports in a deterministic way may easily undermine the cooperative process, they are easier to detect when the attack persists in time: the reports provided by the malicious node will always deviate from those sent by honest users. On the contrary, if sensing data is chosen randomly, the probability of a successful attack diminishes, but also the probability of detection because the malicious node could be confused with a malfunctioning node.

As pointed out in [Chen et al., 2008b], successful detection of forged reports strongly relies on the robustness of the fusion scheme used by the CRN to merge sensing data and take a decision about the existence or absence of primary users.

2.2.2 Network layer

Network layer deals with end-to-end packet delivery and thus provides functionalities such as routing, flow control and QoS. Attacks to routing protocols focus on either directly producing a DoS or, in most cases, modifying the protocol behavior to acquire a profit. Since the routing protocols used in distributed CRNs are the same as in other existing networks, well-known attacks to routing, and especially the ones targeted to ad-hoc routing, also apply here. In the following we describe a short of them.

Selective forwarding attack

A malicious node could discard certain kind of packets avoiding their propagation. With this selective forwarding an attacker could, for example, denial the service to some specific users and/or applications. The selective forwarding attack becomes more powerful when the attacker controls a node with a high routing flow.

Sink-hole attack

In this type of attacks, a malicious node tries to become more "attractive" in terms of routing. As a result, it can force a high flow of data going through it (sink-hole) so that the attacker has control over a high amount of data and can implement any attack to them: DoS , selective forwarding, eavesdropping, etc. When the sink-hole is used to implement a DoS attack is often referred in the literature as black-hole.

Worm-hole attack

A wormhole is an implementation of a hidden tunnel between two points of an ad-hoc network. The wormhole is used to route messages from one point to the other. Obviously, by means of the wormhole, the attacker can force participants to think that they are closer than they actually are, and consequently alter the routing behavior and data flow.

2.2.3 Transport layer

The transport layer is responsible for providing transparent transfer of data between end systems in a reliable manner. TCP [Jacobson, 1988, Allman et al., 1999] is currently the predominant protocol operating at this layer in the Internet and many applications, such as HTTP, e-mail or file transfer, rely on it to successfully deliver their data. However, the lack of security in the design of its mechanisms can be easily exploited by attackers in several ways.

In this section, we describe some of the threats targeted to the TCP protocol focusing in packet injection and cross-layer attacks executed at lower levels. It is worth to emphasize that these attacks are simpler to carry on and more effective in combination with the sink-hole attack, which allows the attacker to eavesdrop packets belonging to a given TCP connection and modify the flow of data.

Packet injection

Packet injection occurs when an entity other than one of the endpoints generates traffic using the source address of an endpoint. By means of it, malicious users can perform several attacks such as DoS, injection of false information or access to resources protected by authentication methods based on IP addresses. In the following, we briefly describe some examples of packet injection attacks targeted to TCP.

In order to perform an attack to a TCP live connection, the IP addresses and port numbers of such connection must be known or at least, successfully guessed. Having this information, the attacker can inject a TCP segment into the connection with the RST bit set, the SYN bit set or false data, by assigning to the forged segment a valid sequence number. If the sequence number is out of the expected range, the segment will be discarded by the receiver with no consequences. On the other hand, with a valid sequence number, the attacker can manage to abort the connection by setting the RST or the SYN bit into the forged segment. Another attack is based on session hijacking, i.e., the attacker intercepts an already established connection between two entities where authentication is needed. As authentication takes place only at the time of the establishment of the connection, the attacker can easily steal the connection and communicate with one of the endpoints bypassing the session authentication.

Cross-layer attacks: from network to transport layer

As it has been previously mentioned, these type of attacks try to disrupt the operations of a particular layer (target point) by exploiting one or more vulnerabilities presented in another layer (attack point). The fact that the attack point and the target point reside in different layers of the protocol stack makes it difficult to detect it by using conventional detection schemes. Note that insider attackers can access all layers, increasing the relevance of these attacks.

An example of cross-layer attacks are the JellyFish (JF) attacks [Aad et al., 2004]. The emergence of the attacks of JF prompted a significant change, since the protocol attack perfectly conforms to all routing and forwarding protocol specifications, making very difficult to distinguish this attack from congestion or packet losses in the network.

The basic idea underlying this attack is to reduce the throughput of all traversing flows which are responsive to network conditions such as delay and loss, e.g., TCP. End-to-end control protocols infer network status from feedback measures and the JF attack interferes with these measures to degrade the traffic flows. JF attacks disrupt TCP connections by means of packet disordering, packet dropping and delay variance.

The first JF variant is a packet reordering attack. TCP has a well-known vulnerability to packet misordering, which generally occur due to factors such as route changes or the use of multi-path routing. The strategy of this attack is to deliver all packets, but once they have been reordered. Although several TCP modifications have been proposed [Blanton and Allman, 2002, Wang and Zhang, 2002] to deal with such problem, none of them are robust to malicious and persistent reordering as the one employed by the JF misordering attack.

The second JF variant is based on periodic packet dropping, according to a maliciously chosen period. In [Kuzmanovic and Knightly, 2006], Kuzmanovic and Knightly show that if losses occur periodically near the Retransmission TimeOut (RTO) value of the connection, its throughput is almost zero. Thus, a JF node does not need dropping more packets than its neighboring congested nodes do, making especially hard to detect the attack.

Finally, the third JF variant consists in packet delaying, but preserving the order of the packets, in order to thwart TCP timers and congestion inferences. The attack strategy consists in waiting a random time before serving each packet, maintaining FIFO order, but significantly increasing delay variance. This fact leads to the expiration of the retransmissions timers and the subsequent reduction of TCP throughput.

Chapter 3

Countermeasures

The effectiveness of the attacks outlined in Chapter 2 depends on several factors such as the attack scenario, the resources of the attacker and the particular attributes of the CRN being compromised. As pointed out by Brown et al. [Brown and Sethi, 2008], the architecture of the CRN strongly determines how vulnerable is the network to these attacks. As an example, a PUE attack can easier deny communication to a CR which performs spectrum sensing on its own than in a cooperative framework. On the other hand, attacks based on intercepting sensitive information are prone to be launched in cooperative CRNs where their member CRs exchange information over common control channels.

The heterogeneity of these networks with regard to underlying technology, architecture, sensing mechanisms, etc, makes hard the design of a global security system suited for every CRN. Traditional approaches used to provide robustness and network security are not adequate for CRNs. Therefore, there is a need for designing effective mechanisms to guarantee its correct operation, taking profit of the cognitive capability of CRs and the cooperative nature of CRNs. In this chapter, we provide an overview of the proposals appeared in the literature in order to counteract such attacks, specially focusing on those which are specific to CRNs, but also providing some general countermeasures for the rest of attacks.

3.1 Jamming countermeasures

The jamming attack has been extensively studied in wireless networks. Existing solutions include physical layer defenses such as directional antennas, spread spectrum, link layer defenses such as frequency hopping and network-layer defenses such as spatial retreats [Wang et al., 2006, Li et al., 2007, Xu et al., 2007]. However, most of them cannot be applied directly to CRNs where spectrum availability varies over time and omnidirectional antennas are often used for communication.

Taking into account the constraints introduced by CRNs, the authors in [Wang et al., 2011] proposed a strategy for secondary users to counteract jamming attacks based on frequency hopping. The target is to optimally select channels for transmitting data and control messages, in order to maximize the throughput of secondary users. With this purpose, the interaction between secondary users and attackers is modeled as a zero-sum stochastic game with two players. Secondary users are considered as a single player, and the second player is an attacker trying to jam as many data and/or control channels as possible. At each state of the game, secondary users analyze different parameters such as channel availability and quality and the attacker's strategy. Based on this information and applying the minimax-Q learning algorithm [Littman, 1994], they can derive the optimal configuration for the network with regard to the selection of channels and its assignment to data or control messages. It is shown that the best allocation strategy consists in assigning more channels for data transmission when the quality of these channels is low, while reserving more channels for sending control messages as quality increases.

Hopping across different bands is also the solution proposed in [Wu et al., 2010]. They provide a model where the players are a single secondary user and m malicious users, which relies on a Markov decision process approach [Puterman, 1994]. At each stage of the game, the attacker is assumed to execute a PUE attack whenever detects any CRN transmission through spectrum sensing. Then, the secondary user decides whether to remain at a given channel or to hop, based on the observation of the current and past slots. A learning algorithm is applied to discover the attacker's strategy and, based on it, derive the optimal defense strategy at each step.

3.2 PUE countermeasures

Protecting CRNs from PUE attacks is indispensable and requires devising robust techniques for verifying the authenticity of primary signals, such as TV broadcast systems or wireless microphones. The simplest way to achieve it would be to embed a signature in an incumbent signal or to use an authentication protocol between primary and secondary users. However, these approaches do not conform to the requirement established by the FCC [FCC, 2003b], which states that *no modification to the incumbent system should be required to accommodate opportunistic use of the spectrum by secondary users.*

Under such demand, most of the proposals appeared in the literature to detect PUE attacks are based on energy measures, i.e., Received Signal Strength (RSS) measures. In CRNs, energy detection is typically used with spectrum sensing purposes. Therefore, it is often the preferred mechanism to detect PUE attacks, as then any extra hardware is required in the CRs.

In [Chen et al., 2008c], the authors propose to use signal energy level detection in order locate a given source and protect CRNs from PUE attacks. This approach is based on the assumptions that primary transmitters have fixed locations and that CRs are devices with limited transmission power. If the position of TV towers is known to the CRN, once a given source is located a verification process is performed in order to distinguish between legitimate primary transmissions and PUE attacks. It must be noted, however, that an attacker could still transmit in the vicinity of a TV tower to get around this method. In such case, the authors assume that the energy level received at the CRN would be much lower than the corresponding to a real TV emitter, and therefore the PUE attack can be effectively detected. One of the drawbacks of this scheme is that relies on the existence of a set of dedicated nodes within the CRN, named Location Verifiers (LVs), which are responsible for performing RSS measures. This fact increases the cost of the CRN deployment and also creates the need for secure exchange information among LVs to avoid potential attacks such as eavesdropping, insertion, modification, or replay attacks.

Chen et al. propose in [Chen et al., 2009b] a method for detecting PUE attacks, also based on the observation of the received signal power, under the hypothesis that an attacker can easily emulate a primary signal but not the specific characteristics of the channel. The authors state that the value of the received power and its variance will vary depending on the source. As a consequence, if these parameters are known for primary users, it is possible to distinguish between a legitimate primary source and a fake one. However, as the authors observed, an attacker can still overcome this countermeasure by predicting the mean and variance of the power measured by a given receiver for a primary user, and transmit a signal with the same characteristics from the receiver's point of view. Another drawback of this method is that it requires a long time of observation to accurately estimate the power values for a particular primary source.

Jin et al [Jin et al., 2009] propose to apply two different types of hypothesis tests, Neyman-Pearson Composite Hypothesis test and Wald's Weighted Sequential Probability Ratio Test (SPRT) to decide whether a given transmission is coming from an attacker o not. As an advantage, this approach does not require the use of a dedicated sensor network or enhancement of the secondary nodes. However, its efficiency is somehow limited since it assumes that the attacker is located within the CRN area, i.e., far away from primary users, and that it does not have power control, i.e., it always transmits at a fixed power.

Energy-based approaches can generally deal with PUE attacks, assuming primary users with known and fixed locations, and with transmission powers considerably higher than an attacker. Despite it, in network environments where primary users are mobile and transmit with low power transmission, i.e. wireless microphones, they are prone to fail in detecting such attacks.

There exist alternative countermeasures such as Radio Frequency Fingerprinting (RFF), which has been widely reported in the literature as a technique for transmitter identification [Ureten and Serinken, 2007, Toonstra and Kinsner, 1996]. It is based on the fact that the radio signal emitted by a radio transmitter when it is activated exhibits a transient behavior with respect to instantaneous frequency and amplitude. Even transmitters of the same type will show different characteristics during a transient period of time due to factors such as age or tolerance levels, which allow to uniquely identify every single transmitter. These transient properties are known as the fingerprint of a radio device. In spite of its complexity, the advantage of this technique is that it can be applied to identify any transmitter, no matter whether it has a fixed location or it is mobile.

In [Afolabi et al., 2009], a pattern recognition technique based on the Electromagnetic Signature (EMS) of the devices is proposed. The method requires recording a wide variety of signals emitted by both CRs and incumbents, extracting the transient portion that reflects the unique attributes of each device for each signal, and storing the different EMSs. An emitter is considered to be an attacker whenever the received signal matches any of the stored CRs signatures which mimic the waveform of an incumbent, or if does not match any of the signatures, which implies that the signal does not come either from a primary user or a CR. The Mahalanobis distance is used to measure how the extracted features of a given signal differ from the stored EMSs and a threshold is applied to decide when there is a match. Thus, the performance of this method strongly depends on the proper setting of the threshold. On the other hand, as the transient behavior of emissions vary due to aging and degradation of the devices, the set of EMSs must be periodically updated in order to be able to detect malicious devices.

With regard to PUE attacks based on wireless microphones, little research has been done until the moment. Besides RFF-based techniques, the methods described above are not applicable because wireless microphones are low power devices with a transmission range of at most 100-150 meters. Anti-PUE schemes based on cooperative energy sensing, where decisions are taken based on measures performed by many CRs, would probably fail on detecting such attack: due to the large dimension of CRNs, most of the CRs would not be in the range of the emitter. Moreover, wireless microphones are usually mobile, so that basic location techniques are not either useful to determine whether a given source is legitimate or not.

Being aware of the difficulty of detecting such attacks and against the requirement established by the FCC, the 802.22 WG proposed the use of a beacon protocol [Buchwald et al., 2008]. In this approach, wireless microphones should send a special signal before starting transmission for signaling their presence. It could be embedded additional information, such as digital signatures, in the beacon signal, thus allowing to easily authenticate the device. In [Lei and Chin, 2008], an improvement to this protocol was presented in order to provide more reliability and balanced error protection for the beacon components. However, the authors in [Chen et al., 2011] state that a great number of legacy microphones still in operation will probably not incorporate the beacon protocol in the near future. As a consequence, they propose a different method which exploits the correlation between RF signals and acoustic information to verify the existence of wireless microphones. Although this method does not require any modification on microphones, it assumes that secondary users are equipped with a sound sensor.

Finally, in [Li and Han, 2010], the authors proposed an approach similar to random frequency hopping, where secondary users randomly select a channel to transmit, avoiding PUE attacks. This method was coined *dogfight in spectrum* due to the similarity of the competition between secondary users and the attacker with evasion games in real air flights. Frequency hopping represents an effective countermeasure to avoid PUE attacks and can also deal with traditional jamming attacks. However, it leads the CRN to continuously perform frequency handoffs, implying the interruption of all communications until the CRN is completely operating at a new frequency. Moreover, the success of this countermeasure strongly depends on the number of available channels when the attack is performed: lack of alternative channels can lead to a DoS .

3.3 Learning engine attacks countermeasures

Attacks targeted to the on-line learning used by CRs devices are based on altering the behavior of the wireless media (by jamming at specific time and frequencies or intentionally transmitting forged signals) with regard to a policy parameter, e.g., the security level. Thus, they modify the learning curve to the attacker's advantage, e.g., achieving a low security level. In order to prevent such attacks, no learning algorithm based on observation of an exposed media should completely trust the collected data. Consequently, the learning curve must be protected in order to avoid achieving unacceptable levels of some parameters. As a naïve solution, threshold values could be defined for every radio parameter, preventing communication when one parameter or a set of them do not fulfill their predefined threshold. In any case, this solution just turns the Learning Engine (LE) attack into a DoS attack but it does not prevent it at all.

Burbank identified four key requirements [Burbank, 2008] that CRs should meet in order to protect the learning process:

- The ability to authenticate local observations.
- Securing data exchanges among CRs used in cooperative mechanisms.
- The ability to authenticate observations received from other CRs.
- The ability to self-analyze their behavior.

As previously mentioned, securing data exchanges can be achieved by using traditional cryptographic primitives. Likewise, anomalous observations coming from malicious or compromised CRs can be identified and excluded from the learning process. In Section 3.5, a more detailed description of these approaches is provided. However, to quantify the reliability of data acquired through direct observation of the media is still challenging.

3.4 Selfish behavior countermeasures

Due to the negative impact of selfish nodes on the cooperative sensing mechanisms of CRNs, appropriate strategies are needed in order to enforce cooperation. In wireless networks, this is a hard problem to deal with, since lack of feedback may be either due to selfishness or information loss. The community research has put a big effort on this topic in the context of ad-hoc networks [Michiardi and Molva, 2003, Milan et al., 2006], mostly regarding routing mechanisms. Although these approaches cannot be directly applied to CRNs, they can serve as a basis to develop efficient methods to detect selfish behaviors and to incentive CRs to cooperate. In addition, these methods should be also tailored to the type of CRN: ad hoc or infrastructure-based. In ad-hoc networks, selfish detection must be performed locally by neighboring nodes and cooperative enforcement may be achieved by traditional mechanisms [Michiardi and Molva, 2003, Milan et al., 2006] which penalized selfish nodes, for example, by not routing packets coming from them. On the other hand, in infrastructure-based networks such as IEEE 802.22, the BS is responsible for gathering the sensing reports from all CRs and therefore should be also in charge of detecting undesired behaviors. Upon request of spectrum sensing, the BS can easily identify which CRs have not reported any measure. As in IEEE 802.22 networks the available bandwidth is distributed by the BS, a naïve approach to enforce cooperation would be to penalized selfish nodes by assigning them a smaller portion of bandwidth or none at all.

As far as we know, the only contribution on this topic was presented in [Song and Zhang, 2009], where the cooperative spectrum sensing process is modeled as an N-player non-cooperative non-zero-sum repeated game. Two different strategies based on game theory are analyzed in order to incentive cooperation: Grim Trigger and Carrot-and-Stick. In Grim Trigger, cooperation from each node is measured and all nodes stop cooperation whenever deviation from any node is detected. This technique can effectively motivate nodes to cooperate but results in poor performance when channel errors are not negligible. On the other hand, Carrot-and-Stick is an iterative approach where, at each step, a given node only cooperates if all nodes cooperated or deviated in the previous step. This approach recovers cooperation when all nodes deviate at the risk of leaving the CRN with no information about spectrum availability. This fact could lead to the usage of spectrum bands occupied by primary users, something that must be avoided in any CRN.

3.5 False feedback countermeasures

In order to deal with false data introduced into the primary detection process , existing proposals rely on fusion schemes capable of detecting anomalies in the reported sensing data and discriminate among malicious users and honest ones. In this way, reports from potential malicious users can be discarded during the fusion process or at least, have a little impact on it. The first work in this direction was presented in [Chen et al., 2008b], where the authors gave an insight of the steps to perform in order to make the cooperative process robust to liars. First, they suggested to employ a fusion scheme different from traditional approaches such as Bayesian detection or Neyman-Pearson test, and proposed SPRT as a candidate to replace them. The SPRT method uses the likelihood ratio as decision variable as in (3.1), where H_0 and H_1 denote the hypothesis that a primary user is absent or present respectively and u_i the observation for the ith user.

$$D_n = \prod_{i=0}^{n} \frac{P\left[u_i|H_1\right]}{P\left[u_i|H_0\right]} \tag{3.1}$$

Contrary to the previously mentioned techniques which use a fixed number of observations samples, SPRT uses as many observations based on need while providing a lower bound for misdetection and false alarm. Indeed, this approach can fit into the cooperative sensing mechanism used by CRNs, where the number of sensing reports may vary. As disadvantages, it requires a priori knowledge of the probabilities of u_is, which may not be known to the CRN and assumes identical probability distributions for all u_is.

Besides, they proposed to apply a reputation scheme similar to those already used to secure routing in ad hoc networks. Up to now, most proposals appeared in the literature have followed these guidelines.

In [Chen et al., 2008a], a Weighted Sequential Probability Ratio Test (WSPRT) was proposed, an extension to SPRT which includes a reputation-based mechanism. In a first step, the reputation for each node is updated: initially is set to zero and it is incremented or decremented by one whenever the local sensing report is consistent with the final decision or not, respectively. Then, a modified version of SPRT testing is applied, where each individual measure is weighted according to the reputation value of the node as shown in (3.2).

$$D_n = \prod_{i=0}^{n} \left(\frac{P\left[u_i|H_1\right]}{P\left[u_i|H_0\right]}\right)^{w_i} \tag{3.2}$$

With this method, unreliable measures are either excluded or their impact on the final decision is diminished. However, this comes at the expenses of requiring a larger number of sensing measures with respect to SPRT.

Other reputation schemes are presented in [Chen et al., 2009a, Qin et al., 2009]. Chen et al. proposed a simple model [Chen et al., 2009a] where the reputation degree of secondary users is set to 0, 0.5 or 1. When the report provided by a given secondary user matches the final decision (taken by a data fusion entity), the reputation degree is increased by 0.5 (or kept to 1 if this was the previous value), or decrease by 0.5 if differs from the final decision (or kept to 0 if it was the previous value).

In [Qin et al., 2009], a reputation system is proposed exhibiting several improvements with regard to [Chen et al., 2008a] and [Chen et al., 2009a]. First, the trust value for a given node is computed by also taking into account its behavior in the past (positive behavior if the decision of the node matches the final decision, negative behavior otherwise). Moreover, a forgetting factor is applied in order to give higher weights to recent behaviors and gradually decrease the influence of older ones. Note that a negative behavior may lead to misdetection of a primary user or to a false alarm, i.e. the final decision is that a primary exists when in fact it does not. Because the former can have more serious consequences, negative behaviors when the final decision was "a primary exists" are more penalized by assigning them lower forgetting factors. Then, the final decision about the existence of the primary is obtained by aggregating the sensing results from the set of CRs, weighted according to the trustworthiness of each secondary user.

In [Min et al., 2010], a WSPRT hypothesis test is also used as data fusion scheme, but the weights assigned to each node are computed according to its statistical importance based on a shadowing correlation model applied to a two-dimensional area. The authors define the shadowing random field of a given point (x, y) as the shadowing gain at unit grid area, i.e. δm x δm, centered at the coordinate (x, y). Then, the covariance of the shadowing random field between two different points (x_i, y_i) and (x_j, y_j) is modeled as in (3.3), with d_{ij} the Euclidean distance between the two nodes, σ the standard deviation of shadow fading. D_{corr} is the decorrelation distance which determines the distance above which the correlation falls to e^{-1} and depends on the environment (urban, suburban...). In suburban areas, it typically takes values between 120m and 200m. From (3.3), an expression for the correlation between pairs of sensing reports is derived.

$$R_p(d_{ij}) = \sigma^2 \cdot e^{-d_{ij}/D_{corr}} \tag{3.3}$$

Abnormal reports are detected by analyzing their similarity (or difference) with reports from neighboring nodes. To this end, the set of CRs is divided into clusters according to their proximity so that nodes belonging to the same cluster should be highly correlated. The fact that shadowing correlation decays exponentially with distance implies that, in order to reliably compare measures from different nodes belonging to the same cluster, they should be pretty close, on the order of a hundred of meters. This could represent a serious problem in CRNs where users may be scattered over a large area. However, as mentioned by the authors, some studies [Sawada et al., 2006] suggest that households in rural areas (the scenario where IEEE 802.22 networks are expected to be deployed) tend to be clustered.

Finally, in [Kaligineedi et al., 2008, Kaligineedi et al., 2010], different schemes for detecting malicious users based on outliers detection are evaluated. In statistics, an outliers is an observation that is numerically distant from the rest of the data. When applied to cooperative spectrum sensing, identification of outliers allows discarding those energy measures which are highly improbable, and take a more reliable final decision about the existence of a primary. In [Kaligineedi et al., 2008], an outliers factor is assigned to each secondary user at the kth sensing iteration by using the sample mean and the sample standard deviation of the reported energy values as in (3.4), where $\sigma_n[k]$ is the outliers factor, $e_n[k]$ represents the reported energy value of a given secondary user, and $\mu[k]$ and $\sigma[k]$ are the sample mean and the sample standard deviation respectively.

$$\sigma_n[k] = \frac{e_n[k] - \mu[k]}{\sigma[k]} \tag{3.4}$$

However, as the authors pointed out in [Kaligineedi et al., 2010], this technique does not perform well when the distribution of the energy values is highly skewed. In such case, valid values lying on the heavy-tailed side of the distribution will be assigned a high outliers factor, meaning that they will have a very low impact on the final decision. Because of this, they proposed and evaluated alternative techniques based on other statistical parameters such as the Median Absolute Deviation (MAD) or the Bi-Weight Scale (BWS), which are shown to be more robust than the mean and the standard deviation. The MAD measures the median of the absolute distances of the data points from the sample median as in (3.5).

$$\sigma_n[k] = median_n \left\{ |e_n[k] - \mu[k]| \right\} \tag{3.5}$$

According to [Kaligineedi et al., 2010], the BWS is a more efficient measure of scale than MAD which only discards extreme data points, and is given by (3.6).

$$\sigma_n[k] = \sqrt{(}\frac{N \sum_{u_n^2<1} \left(e_n[k] - \mu[k]\right)^2 \left(1 - u_n^2\right)^4}{s\left(-1 + s\right)}) \tag{3.6}$$

where

$$s = \sum_{u_n^2<1} \left(1 - u_n^2\right)\left(1 - 5u_n^2\right) \tag{3.7}$$

and

$$u_n = \frac{e_n[k] - \mu[k]}{median_n \left\{ |e_n[k] - \mu[k]| \right\}} \tag{3.8}$$

with c_2 a tuning constant which can be used the impact of extreme points on BWS estimates.

3.6 General countermeasures

One of the main characteristics of many CRNs is the cooperation of the participants in order to find spectrum opportunities. The information exchanged with such purpose often carries shared data and thus unicast communications may be replaced with broadcast or multicast communications. Therefore, there is a need for protecting such data against outsiders of the network. As already mentioned in the beginning of the present chapter, it is essential to provide security services such as confidentiality of the communication channel, authentication of the peers involved in an information exchange, and integrity of the messages, among others. This can be achieved by means of cryptographic primitives such as Symmetric Key Encryption (SKE), hash functions and Public Key Cryptography (PKC). Despite it, an outsider could still impersonate a CRN member by replaying old messages already broadcasted by such member.

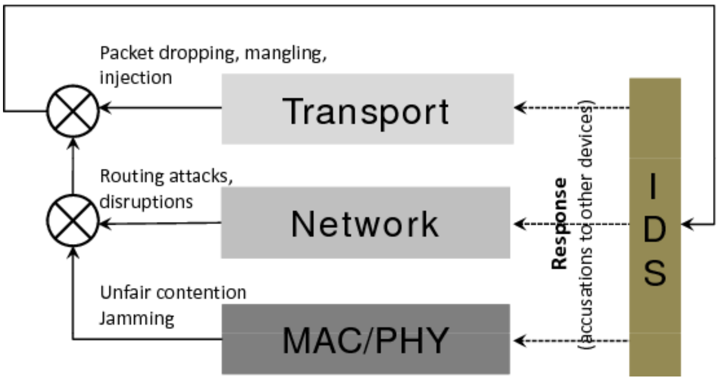

Packet dropping, mangling, injection
Transport
Routing attacks, disruptions
Network
Unfair contention
Jamming
MAC/PHY
Response
(accusations to other devices)
IDS

Figure 3.1: Cross-Layer IDS

As a consequence, to guarantee data freshness through the use of sequence numbers, nonces or other mechanisms also becomes a must.

The use of cryptographic primitives requires the distribution of keying material among the members of the networks. One of the most efficient approaches to deal with such task in broadcast networks is what is called group security. Group security is, thus, targeted to provide group privacy and group authentication: data is protected from outsiders and the only sources of communication are the members of the group. As a result, it is merely based on the use of a common shared secret called the session key or group key. This key allows every group member to: 1) send encrypted data; 2) decrypt received data, and 3) authenticate itself as a group member since the knowledge of the session key guarantees that it belongs to the group. As only the current group members ought to know the session key, such key must be updated every time the membership of the group changes. Group Key Management (GKM) studies the generation and updating of the keying material used for securing the group during its whole life [Wallner et al., 1998]. For sure, the application of known GKM techniques to CRN will secure the exchanged critical data while incurring a low impact to the network performance. In addition, the study of distributed and autonomous GKM protocols [Hernández-Serrano et al., 2008] is necessary in order to cover all the CRN architectures.

Furthermore, we believe that the use of Intrusion Detection System (IDS)s could increase detection capability for a wide variety of attacks. IDSs in CRNs should monitor other devices for intentional deviation from protocol, i.e. misbehavior, detecting which nodes are suspicious or malicious. Traffic monitoring should be done in a distributed and cooperative way because feedback from many CR devices can enhance the efficiency of IDSs. As stated in [Zhang and Lee, 2000], an architecture for better intrusion detection for wireless ad-hoc networks should be distributed and cooperative. To our understanding, the same assertion applies to CRNs where cooperation is inherent to their nature. On the other hand, interaction among layers should be implemented to guarantee the proper detection of cross-layer attacks. Several IDS approaches [Zhang and Lee, 2000, Mishra et al., 2004, Bhuse and Gupta, 2006] fulfill such requirements but their particularization to CRNs is still challenging.

The state-of-the-art presents IDS approaches based on two main techniques: misuse and anomaly detection [Lee et al., 1999]. In the former, a specific model or pattern, known as the "signature", is built for every known attack. The second approach relies on the basic principle that every deviation on the normal execution of the protocol is suspicious.

CRNs comprise a wide variety of technologies and protocols and thus they are exposed to many potential attacks. As a result, IDSs based on misuse detection techniques may be impractical for CRNs due to the difficulties of specifying, distributing and updating signatures of so many attacks. On the other hand, anomaly detection techniques could be applied but are usually prone to high false positive rates that are increased due to the risky environmental conditions of CRNs. As a consequence, new schemes that help in accurate diagnosis of malicious attacks in CRNs are needed. These schemes should employ cross-layer interactions based on observations at several layers to detect new attacks, and decrease the number of false positives. Specifically, it is necessary to define a set of features that correlates information from different layers in order to profile the normal behavior of nodes, as shown in Figure 3.1.

In CRNs, observations from different CRs may vary depending on their location. As a result, channel degradation can prevent a monitor node from observing the media during a certain period of time. This fact leads to initially trust a node asserting that it cannot monitor but opens doors to selfish or cheating behavior. Moreover, a malicious node could intentionally forge information in order to disrupt the overall network performance. For this reason, a reputation system is required to develop an efficient trust model [Mejia et al., 2009] for CRNs. This model should benefit from the redundancy of the network since the feedback of many participants can lead to an easier detection of the attack source. In addition, a global analysis of the live monitoring mechanisms must be implemented in order to observe whether the nodes are cooperating or not and dynamically set trust levels for the collected data. In this sense, the reputation systems described in Section 3.5 could be also applied to improve robustness of other cooperative mechanisms beyond spectrum sensing.

Part III

Contributions to the Security of CRNs

Chapter 4

Roadmap for the Security of CRNs

Although research on CRNs has been object of a big effort, it is still a hot topic requiring further work, particularly with regard to network security. As for any other network scenario, security is usually split into two lines of defense. The first one is focused on avoiding attacks and it is closely related to the use of cryptographic primitives. The second one should be more devoted to detect and identify the attacks that have passed over the first line, so that appropriate preventive actions can take place [Bace, 2000]. Thus, it involves capturing data and looking for evidences to determine whether the system is under attack.

We assume that cryptographic solutions based on PKC, SKE, etc., which have been proposed in the literature for other scenarios, can be successfully implemented in CRNs without significantly changes. For this reason, in this work we will exclusively focus on the design of a second line of defense.

In the literature, it has been widely pointed out the need for a second line of defense in wireless networks, especially in the context of IDSs [Zhang and Lee, 2000, Bace, 2000, Zhang et al., 2003]. As previously mentioned, wireless links open doors to adversaries and are susceptible to many attacks ranging from passive eavesdropping to active interfering. Furthermore, when such networks are decentralized, e.g., ad hoc networks, attack detection becomes an even more difficult task. First, nodes are autonomous units and therefore easier to be compromised and more difficult to control. On the other hand, in such networks decisions are taken in a distributed manner and typically rely on the cooperation of all nodes, making these mechanisms vulnerable to a wide variety of attacks beyond those common to all wireless networks.

CRNs can adopt different architectures either infrastructure-based or decentralized and share many attributes with other wireless technologies. However, as explained in Chapter 2, they exhibit different particularities with regard to attacks vulnerability and therefore, existing proposals addressed to detect attacks in conventional wireless networks cannot be directly applied. Despite it, they should be considered as a starting point for the design of a second line of defense for CRNs.

The work presented in this chapter is intended to be used as a roadmap for securing CRNs, i.e. to provide an overview of those mechanisms needed to efficiently detect and counteract the main attacks to CRNs, either centralized or decentralized. With such purpose, we outline the modules required by a detection system acting as a second line of defense in CRNs, and provide guidelines in order to develop them. As we will see further in Chapters 5, 6 and 7, such guidelines have been followed in order to develop the different contributions of this thesis.

The structure of this chapter is as follows. In Section 4.1 we identify the requirements and main concepts with regard to the implementation of an attacks detection system for CRNs. Next, in Section 4.2 we provide a high-level description of the structure of such system and the tasks to be performed by each of its components. Finally, in Section 4.3 we outline the next steps which have been followed regarding the development of this system and that will be presented in the remaining chapters.

4.1 Securing CRNs

As seen in Chapter 2, there are multiple types of attacks that can affect the performance and integrity of CRNs. The design of a second line of defense for counteracting such attacks requires the definition of several characteristics, such as the sources of audit data, the attacker model and the methodology to use for each type of attack. Besides, the system implementing this second line of defense has its own issues that must be considered, such as the existence of attacks against the system itself or the distribution of the detection entities. These issues have been studied by the state of the art on this area [Mishra et al., 2004], and will be explained in the next sections.

4.1.1 Attacker Model

A key step in the design of the system is to accurately define the attacker model that specifies the capabilities of the adversaries targeting the services of CRNs. As for the attacks it can perform, we have focused on those exclusive of CRNs which can severely degrade their performance: the PUE attack, False Feedback (FF) attacks and LE attacks. Although jamming attacks are not specific to CRNs, they are also considered in this design since without the proper protection mechanisms they can derive into a PUE attack.

With regard to the execution of the attacks, we assume that the knowledge of the attacker can be quite diverse. It can know nothing about the structure of the CRN, trying to learn about it by eavesdropping or implementing some known fuzzy logic techniques [Clancy and Goergen, 2008]. Or it can have complete knowledge about the CRN operation, which enables it to perform sophisticated attacks such as LE attacks. Regarding its transmission power, we will assume that most attackers will make use of small radios with a limited action range, but we will not discard the existence of powerful emitters with the capacity of faithfully emulating a primary TV signal. As for the number of devices owned by the attacker, it could range from one till many cooperating radio devices, which could difficult the operation of the detection mechanisms. As the mobility of the attacker can degrade the functionality of the system, we will also assume that the attacker can both move within a given area or remain at a fixed position. Finally, concerning FF attacks, we will consider malicious nodes providing both random false measures and deterministic ones in order to maximize the damage caused to the CRN mechanisms.

4.1.2 Detection entities and general architecture

With respect to the sources of audit data, each CR can take the role of monitoring the data it manages and its surroundings. Besides, as CRs are not expected to be constrained devices, such as nodes in Wireless Sensor Networks (WSNs), they could afford the existence of a detection entity in every CR. In this way, detection mechanisms can be performed by a CR in an autonomous manner based on its own perception of the environment. If the evidence is inconclusive or there is a need to have a holistic point of view of the situation, the CR can make use of the distributed nature of the network and use a collaborative mechanism to take a given detection action. Even in centralized CRNs, the existence of distributed detection entities can help to develop a more efficient system and relieve the BS of some burden. However, in such networks, the BS may exclusively take the role of detection entity gathering the reports provided by the set of CRs, and leave CRs as merely sources of data.

Regardless of their distribution, detection entities should provide some cross-layer functionality where information from different network layers are used as an input to the components, so as to improve attack detection rate.

After these issues have been discussed, it is necessary to describe the potential architecture of the system. In fact, there is a “de-facto” agreement on the basic elements of an attack detection system [Giannetsos et al., 2009]: a local packet monitoring module that receives the packets from the neighborhood, a statistics module that stores the information derived from the packets and information regarding the neighborhood, a local detection module that detects the existence of the different attacks, an alert database that stores information about possible attacks, a cooperative detection module that collaborate with other detection entities located within the neighborhood, and a local response module that take decisions according to the output of the detection modules.

4.1.3 Detection methodology

The detection modules outlined in the previous section must make use of first-hand information, i.e. information acquired by the entity on its own, second-hand information obtained from other entities, and statistical data. Then, these modules can use such data in order to distinguish between normal and abnormal activities, thus discovering the existence of intrusions.

There are actually three main techniques that can be used to classify actions: misuse detection, anomaly detection, and specification-based detection [Axelsson, 2000]. The first technique compares the collected information with predefined “signatures” of well-known attacks. The second technique store patterns of what can be considered as “normal” behavior. These patterns are built by means of machine learning techniques and training, and the system react against any significant deviation of those patterns. Finally, the third technique is also based on deviations from normal behavior, although the concept of normal behavior is based on manually defined specifications.

Misuse detection techniques are very effective identifying instances of the known attacks but have no ability to detect new ones. On the other hand, as mentioned above, the rules used by anomaly detection schemes are established based on learning techniques. Therefore, they may be more effective than specification-based techniques in dealing with unexpected situations, not considered in the design phase. However, they are also prone to be disrupted by an attacker which interferes the learning process, if it is not performed in a secure environment.

From the previous reasoning, we believe that a detection system for CRNs should combine these three techniques. By way of illustration, a signature-based scheme could be used to detect PUE attacks based on wireless microphones, by defining a signature rule as ”there is a wireless microphone transmission coming from the same device every time the CRN switches to a new channel”. On the other hand, anomaly-based or specification-based techniques could be applied to detect LE attacks. As a naïve example, let us consider as ”normal behavior”, the fact of achieving high throughput when the CRN is operating on a channel with good environment conditions, e.g. high SNR, regardless of the level of security being used. If an attacker jams the channel only when a high level of security is set, the throughput will decrease and a deviation from the expected behavior will be clearly identified.

Based on the application of these techniques, the system should provide with state information on the different attacks. For example, by means of a level of confidence such as ”with p% confidence, attack X has been performed ”. The confidence level should be set according to how close is an action to any of the predefined signatures, or how network performance differs from the expected behavior.

4.1.4 General Requirements

When designing the blueprint of the second line of defense and the functionality of its detection modules, it is necessary to consider certain requirements that its elements must fulfill. These are the most relevant [Mishra et al., 2004]:

- The system must not introduce new weaknesses into the system. For example, the cooperative detection module must take into account the existence of malicious and faulty nodes, and the existence of DoS attacks targeting the message management systems must be prevented.

- The system must be fault-tolerant, able to run continuously and recover from problematic situations. The existence of mechanisms that store the current and previous states of the network must be considered in the design.

- The system must provide adequate mechanisms that allow users or the network itself to know about the existence of a certain attack and react against it. This includes attacks against the system itself.
- The design of the system must allow the addition of new detection modules, or a seamless interaction with existing detection mechanisms. Note that any detection module must be as accurate as possible, with fewer false positives and false negatives.

4.2 Blueprint of a detection system for CRNs

In this section we will define the blueprint of a second line of defense for CRNs, which will contain the different detection mechanisms for the attacks outlined in Section 4.1.1. Such blueprint can be used as a foundation for the creation of a functional and usable detection system. This system is intended to be implemented in an entity gathering data provided by a set of CRs, which is assumed to have enough resources with regard to memory and computational capacity.

The architecture of the proposed system is shown in Figure 4.1 and includes the following modules: input, memory, output, and detection. The input module is in charge of managing the first-hand information, the second-hand information, and the cooperative processes. The memory module is used to store the statistical information derived from the input and to provide an interface to the specific network information managed by the CR. The output module takes decisions according to the output of the detection modules, e.g., it informs the user or a central system, and stores other information such as the alert database. Finally, the detection module detects the existence of the different attacks using as an input the data provided by the input and memory modules. Figure 4.1 sketches the different modules with their relationships. Note that this blueprint is not exclusive, as it can be possible to add new detection mechanisms that will take advantage of the existence of the input, memory, and output modules. Next, we focus on the modules composing the detection module.

4.2.1 Module of Cooperative Location of Primary Emitters

As mentioned in Section 3.2, locating potential primary transmissions may lead to mitigate or at least effectively react against PUE attacks. Regarding physical location of RF transmission sources, most of the proposals appeared in the literature rely on measures of certain distance dependent parameters performed at the BS or at nodes whose position is well known. Typically, these parameters are [Patwari et al., 2005]: 1) RSS, the signal strength received at a particular node; 2) Angle of Arrival (AoA), which determines the direction of arrival of an incident signal,

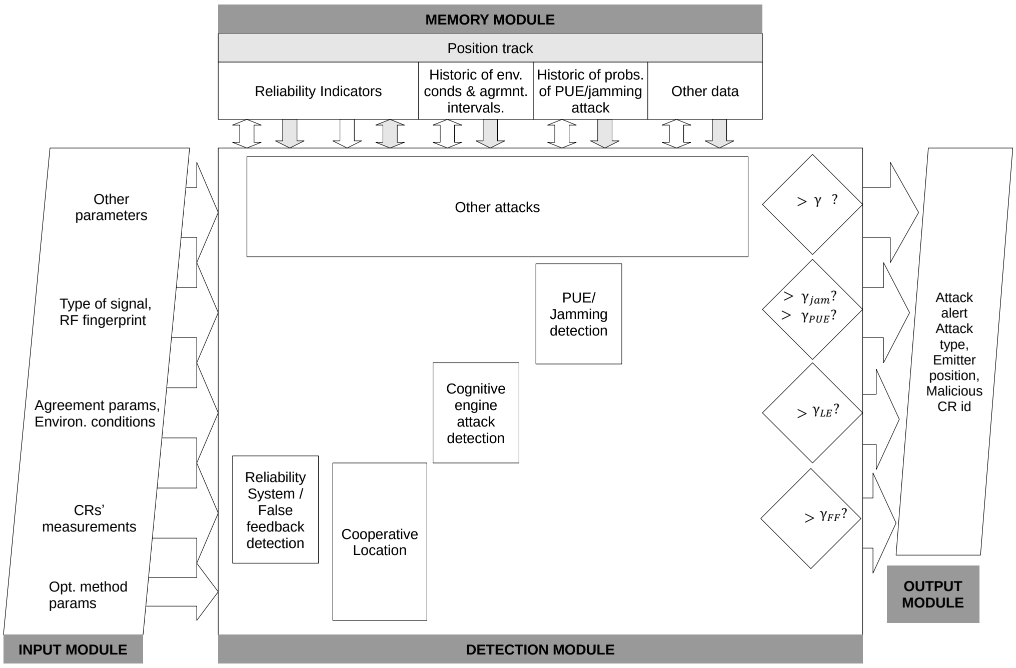

MEMORY MODULE
Position track
Reliability Indicators
Historic of env. conds & agrmnt. intervals.
Historic of probs. of PUE/jamming attack
Other data
Other parameters
Type of signal, RF fingerprint
Agreement params, Environ. conditions
CRs' measurements
Opt. method params
INPUT MODULE
Other attacks
PUE/ Jamming detection
Cognitive engine attack detection
Reliability System / False feedback detection
Cooperative Location
DETECTION MODULE
$> \gamma$?
$> \gamma_{jam}$?
$> \gamma_{PUE}$?
$> \gamma_{LE}$?
$> \gamma_{FF}$?
Attack alert Attack type, Emitter position, Malicious CR id
OUTPUT MODULE

Figure 4.1: Modules of an attacks detection system for CRNs

3) Time of Arrival (ToA), the time taken by the signal to travel between two nodes, and 4) TDoA, the time of arrival difference between pairs of nodes.

A CRN can take advantage of its cooperative nature by collecting measures from the set of CRs and based on them, estimate the position of the emitter. However, these measures are usually affected by several sources of error, such as shadowing, multipath or performance limitations of the device itself. This problem can be partially overcome by considering a larger set of measures. Besides, location algorithms such as Taylor series estimation [Foy, 1976] or the Kalman-Bucy filter [Kalman and Bucy, 1961] can be applied in order to minimize the error performed in the position estimation .

We represent the cooperative location module as in Figure 4.1, with the following inputs and outputs:

From input module:

- Range/angle measures from the set of CRs.
- Input parameters for the location algorithm. As an example, some algorithms such as Taylor-series estimation require an initial guess or the measures error model. On the other hand, Kalman-Bucy is typically used to locate mobiles nodes and requires, in addition, a mobility pattern.

From memory module:

- An indicator of the reliability of the measures based on previous results and computed by the reliability system module, which is described in Section 4.2.2.

- Read access to the previously stored emitter's position and its associated estimation error.

To memory module:

- The estimated position of the primary emitter. The estimation position should follow the format $f(t) = (x, y, z)$ being t the time axis. Note that $f(t)$ is constant when the primary source is static.
- A guess of the error performed in the estimation of the position.

4.2.2 Module of Reliability System and False Feedback detection

This module is in charge of measuring the reliability of the range/angle measures provided by a given CR. According to the emitter's position given by the cooperative location module, it computes the likeliness of a node reporting a given measure. The lower the probability of reporting the current measure, the less reliability is assigned to that node's measures. Note that there may be differences between the expected value and the measure reported by a given CR due to either environment conditions, e.g., shadowing or multipath, or due to misbehavior of the node. The module should be able to distinguish between both cases, for example, by correlating measures provided by neighboring nodes which are expected to report similar values. Besides, the module should take into account past measures provided by the same node to compute its reliability. In this way, malicious nodes which repeatedly provide false feedback can be identified, as its reliability will progressively decrease.

From input module:

- Measures reported by the set of CRs and the node implementing the module (typically the BS in centralized CRNs).

From memory module:

- Previous reliability indicators for the set of CRs.
- The estimated position of the primary emitter obtained by means of the cooperative location method and the corresponding estimation error.

To memory module:

- Updated reliability indicators for the set of cooperating nodes reporting measures.

To output module:

- The module outputs an alert of FF attack and the identification of the suspicious node/s whenever the probability of FF attack exceeds a given threshold γ_{FF} . The probability of FF attack for a given node should be computed based on its reliability indicator. For example, let us denote the reliability indicator for node i as $r_i \in [0,1]$. Then, the probability of this node performing a FF attack could be computed as $p^i_{ff} = 1 - r_i$.

4.2.3 Module for Detecting Jamming and PUE Attacks

Jamming attacks interfere with the channel used for the CRN operation, thus degrading its communications. Besides, it may force the network to switch to another channel with better conditions. If the attack is repeated whenever the CRN switch to a new channel, the throughput can be dramatically reduced and even lead to a DoS. PUE attacks have the same purpose but differ from simple jamming in that they emulate primary transmissions instead of just producing interferences. Because of this, they are harder to detect and may have a more harmful impact on CRNs.

In IEEE 802.22 networks, PUE attacks can be classified depending on the type of primary signal into TV signal-based and wireless microphone-based attacks. Attacks based on jamming or wireless microphone-based PUE may be detected with an anomaly detection scheme: jamming or wireless microphone transmissions appearing whenever the CRN switches from one channel to another. As a result, this module should be able to identify an attacker "chasing" the CRN. This could be achieved by looking for matches between the RFF signature and the position of the current emitter and those stored at the memory module for previous attacks. In the case of mobile attackers, specific location algorithms such as the Kalman-Bucy filter should be applied. By feeding such algorithms with a mobility pattern, a relation between the current position of the emitter and those kept in the database could be established.

On the other hand, TV signal-based PUE attacks can be more easily overcome since legitimate TV primary emitters' positions are assumed to be fixed and known. Because of this, PUE detection could be performed by comparing the estimated position given by the cooperative location module with those stored at the database of TV emitters. RFF techniques [Ureten and Serinken, 2007] could also be used in order to improve the rate of false positives/negatives.

From input module:

- Type of signal: interfering signal or primary signal (e.g. TV or wireless microphone signals). Any signal with different characteristics to those of a primary emission, e.g., TV or wireless microphone transmissions, should be consider as jamming. Mechanisms for detecting primary signals have been widely studied [Zhengyi et al., 2010, Nieminen et al., 2010].
- RFF of the emitter.

From memory module:

- Position estimation of the emitter computed by the cooperative location method and its associated error.

- RFF of previous jamming/PUE attackers.
- Probability of jamming/PUE attack for the current emitter, if it has been previously identified as an attacker. A given emitter can be identified by means of its position and its RFF.

To memory module:

- Updated probability of jamming/PUE attack for the current emitter.

To output module:

- If the probability of being under a jamming attack is above a certain threshold γ_{jam}, the module outputs an alert of jamming attack by the current emitter.
- If the probability of being under a PUE attack exceeds a certain threshold γ_{PUE}, the module outputs an alert of PUE attack by the current emitter.
- If an alert of either a PUE attack or jamming attack is generated, the module outputs the estimated position of the emitter and its associated error.

4.2.4 Module for Detecting Attacks to the Learning Engine

The basis of this module is to detect abnormal environment conditions with regard to the use of some transmission parameters such as security, modulation, codification, etc. Consequently, it should take as input statistics on the environment characteristics and network configuration profiles being in use. Based on such information, it should look for large deviations from the expected network performance. As discussed in Chapter 2, many LE attacks may be performed when transmission parameters are being negotiated. As a consequence, correlation between agreement time intervals and abnormal bad environment conditions could provide a given probability of being under an LE attack. If this probability becomes greater than a certain threshold γ_{LE} an alert should be generated.

From input module:

- Boolean indicating if there is an on-going parameter agreement and, being the case, the parameters in negotiation.
- Current environment conditions.

From memory module:

- Historic of environment conditions, i.e., normal values, variance, predictions, etc. and agreement time intervals.

To memory module:

- New environment data and agreement time interval.

To output module:

- If the probability of being under a LE attack is above a certain threshold γ_{LE}, the module outputs an alert of LE attack, the parameters under attack and the estimation of the emitter's position, if available.

4.2.5 Additional modules

The modules previously described deal with the main attacks to CRNs, such as the PUE attack or LE attacks. Despite it, as CR is a relatively new technology, new vulnerabilities will be discovered and malicious users will probably improve their techniques in order to overcome this second line of defense. Because of this, the proposed system must be flexible in order to include new modules to identify and counteract upcoming threats. These modules should be designed following the same principles as the ones outlined in Sections 4.1 and 4.2, i.e., combining anomaly-based and signature-based schemes, but tailored to the particular attributes of the attack itself. Besides, as pointed out in Section 3.6, data provided by different layers should be correlated in order to increase detection probability for any type of attack, and in particular for cross-layer attacks. By way of illustration, consider the Jellyfish attack described in Section 2.2.3, which is performed at the network layer and is targeted to disrupt TCP connections. A signature for this attack should combine both network and transport layer patterns, i.e., low throughput for a given connection TCP with respect to other flows and no signals of congestion for the links involved in the connection.

4.3 Roadmap for the security of CRNs

In this chapter, we have provided a set of guidelines to implement a valuable detection system for CRNs. Its design is intended to provide a container where the modules previously described can be implemented inside a device and interact with existing interfaces, e.g., the information stored inside the CR, the output of the different network layers. The proposed high-level scheme fulfills the standard requirements for an attack detection system [Mishra et al., 2004] and inherits the "cognitive" behavior of CRNs, which implies learning from the past, making intelligent decisions and positively evolving. Mainly, we have focused on defining the necessary inputs (input module), the storage requirements (memory module) and the attack alerts (output module) generated based on "cognitive" decisions (detection module) from present and past data.

The main target of this thesis is precisely to go a step forward beyond the high-level description of this system. With this purpose, the remaining chapters are devoted to detail the operation of some of its modules and to evaluate its efficiency in detecting the main attacks to CRNs.

In Chapter 5, we propose a new cross-layer attack to CRNs which requires the design of an additional module in order to be detected . As we will further explain, this module makes use of cross-layer mechanisms which take as input data from the link layer and the transport layer. Besides, the cooperative location module presented in Section 4.2.1 can improve the probability of detection of this attack and provide the position of an attacker in order to take the proper actions.

The particular features of the location module are described in Chapter 6. The proposed module outputs the position estimation of a given emitter based on TDoA measures derived from the data provided by a set of CRs. This estimation is used by the PUE detection module in order to determine whether a given emitter is a legitimate primary transmitter or an attacker. In this thesis, we have focused on the capabilities of this module in order to detect PUE attacks based on primary emitters with fixed and known positions, e.g. TV towers. The components of this module regarding detection of PUE attacks based on wireless microphones and mobile attackers will be part of future work.

Finally, Chapter 7 deals with the challenge of designing the reliability module. The proposed implementation detects the existence of false reports provided by malicious or compromised CRs. Besides, it assigns a given weight to each measure according to its reliability, which is then used by the cooperative location module in order to obtain a more accurate position estimation.

Chapter 5

A New Attack to TCP in CRNs

Security in CRNs is a topic requiring further effort from the scientific community. Besides the design of new mechanisms to counteract those threats already identified in the literature, it is needed to envision new vulnerabilities which CRNs may exhibit. As outlined in previous chapters, these networks open doors to attackers not only because of their wireless nature but also due to the specific mechanisms used in their normal operation. In particular, special focus should be put on cross-layer attacks. Because such attacks are performed at a given layer but targeted to a different one, they are hard to detect and can represent a serious threat.

In this chapter we propose a new cross-layer attack to TCP in CRNs, which we have coined the Lion attack. The attack is performed at the physical layer and targeted to degrade the throughput of TCP connections. It is based on forcing a frequency handoff in a CRN, thus interrupting all communications established in the network. Due to the congestion control mechanisms of TCP, these interruptions have a harmful impact on its performance, much higher than in other protocols and can even lead to a DoS. Besides, a set of countermeasures are provided in order to detect and mitigate the effect of the attack. These countermeasures are intended to be included in the security system described in Chapter 4.3. Because this threat has not been identified in previous works, its detection module should be included in the other attacks module of the system, which is gray-shaded in Figure 5.1.

The chapter is structured as follows. Section 5.1 gives an overview of the main features of the TCP protocol in order to fully understand the basis of the attack. In Section 5.2, we provide a description of the attack and the resources needed by the attacker in order to perform it. Section 5.3 presents the analytical model which has been derived for the attack. Section 5.4 is devoted to analyze the impact of the attack both via simulation and by means of the analytical model. In Section 5.5 we propose several countermeasures in order to mitigate the effect of the attack and to detect it. Finally, in Section 5.6 we provide the conclusions of this work.

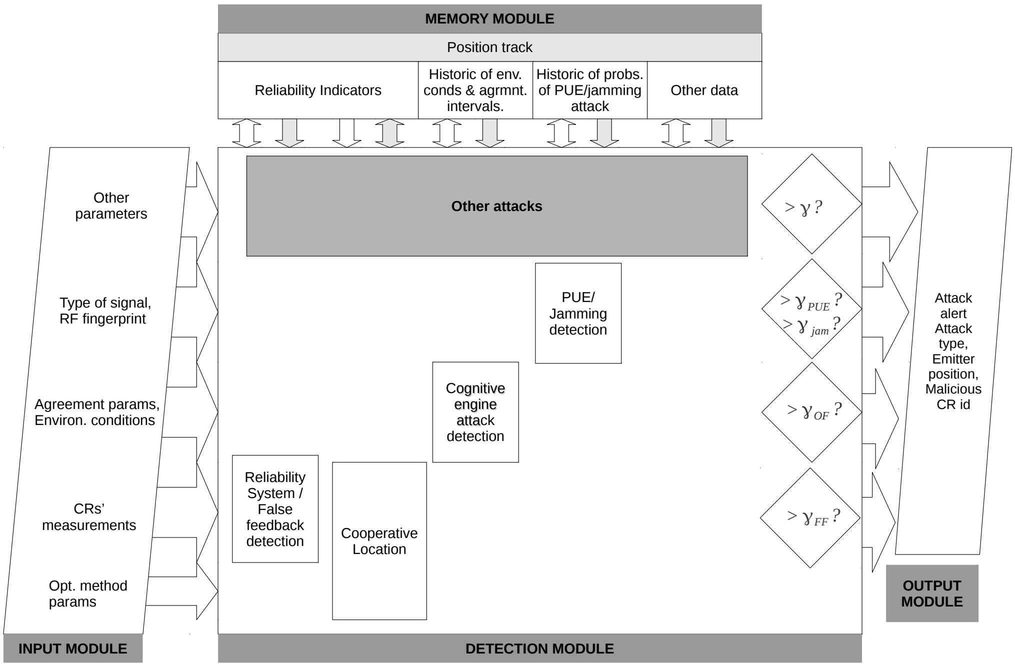


Figure 5.1: A new module to detect Lion attacks

5.1 Background on TCP

The TCP protocol [Jacobson, 1988] is the most widely used transport control protocol on the Internet. The reason for that is because it is a connection-oriented protocol which ensures delivery of data, without duplicates and in a timely fashion. In this regard, the key to providing reliability is that all units of data transmitted in a TCP connection, known as segments, must be acknowledged by the receiver. A retransmission timer is triggered whenever a TCP segment is sent, so that if the timer expires and the corresponding Acknowledgment (ACK) has not been received, the segment is considered to be lost and it is retransmitted.

A sliding window scheme is used to control the amount of data to be sent by a TCP entity. The size of the transmission window indicates the amount of outstanding data that a TCP sender can have unacknowledged at a given point in time, and it is controlled both by flow control and congestion control mechanisms. Flow control mechanisms aim at controlling the flow of data between two devices so that the receiver is not overwhelmed. Based on the feedback provided by the receiver through the ACKs, the value of the receiver window (rwnd) is set, which represents the amount of data that the receiver is willing to receive. Congestion control mechanisms are targeted to adapt the TCP transmission rate in order to response to network congestion. Through these mechanisms, the value of the congestion window (cwnd) is computed, which represents the amount of data that the network can absorbed. Based on them, the TCP transmission window (twnd) is computed as twnd=min(rwnd,cwnd), meaning that the transmission rate is limited either by the capacity of the receiver or by the level of congestion of the network.

5.1.1 Congestion control algorithms

As for the congestion control mechanisms used by the protocol, earlier versions of TCP included very little information about how to handle congestion situations. The TCP Tahoe proposed by Jacobson in 1988 [Jacobson, 1988] introduced the Slow-Start, Congestion Avoidance and Fast Retransmit algorithms. Nowadays, most implementations make use of the algorithms defined in the RFC 2581 [Allman et al., 1999], a version known as TCP Reno, which improves TCP Tahoe by adding the Fast Recovery algorithm. Next sections provide a summary of these four algorithms.

Slow Start

This algorithm is intended to probe the state of the network at the beginning of a TCP connection. It restricts the cwnd to only an amount of data equal to one “full-sized” segment, that is, equal to the Maximum Segment Size (MSS) value for the connection. For each received ACK, the cwnd is increased by the size of another full-sized segment. This is roughly equivalent to double the cwnd every Round-Trip Time (RTT) seconds, meaning that the cwnd grows exponentially. The cwnd is progressively increased until a loss is detected or until the cwnd reaches a predetermined threshold, known as slow start threshold (ssthresh). This parameter, which is initially set to an arbitrary value (typically 65535 bytes), defines the value of the cwnd above which the transmission rate of the TCP sender can lead to congestion. In such case, the TCP sender enters the Congestion Avoidance algorithm mode of operation. On the other hand, when a loss is detected due to the expiration of a retransmission timer three actions take place: the lost segment is retransmitted, ssthresh is reduced to half the value of twnd, and the cwnd value is dropped to one segment. After a RTO occurs, the cwnd is always smaller than ssthresh and therefore the Slow Start algorithm is applied again.

Congestion Avoidance

When a TCP connection is approaching a potential situation of congestion, i.e., the cwnd exceeds the value of ssthresh, the Congestion Avoidance algorithm is applied in order to slow down the increase of TCP transmission rate. For each received segment, cwnd is updated as in (5.1), where cwnd is expressed in bytes. In this way the cwnd is increased roughly by one full-sized segment every RTT s, implying a linear growth of cwnd with time.

$$\mathrm{cwnd} = \mathrm{cwnd} + \frac{\mathrm{MSS} \cdot \mathrm{MSS}}{\mathrm{cwnd}} \tag{5.1}$$

Fast Retransmit

A segment loss can be detected through the expiration of its retransmission timer but this process may take several seconds depending on the value of the RTO. Fast Retransmit is based on the assumption that, if several duplicate ACKs are received, i.e., different segments with the same acknowledgement number, it is a clear evidence of segment loss. Thus, the segment is retransmitted after the third duplicate acknowledgement is received and without waiting for the expiration of the timer.

Fast Recovery

This algorithm is intended to improve TCP performance after a segment has been retransmitted by means of Fast Retransmit. Instead of reducing cwnd to one segment and going back to Slow Start, as it occurs when a retransmission timer expires, the sender resumes transmission with a larger window and enters Congestion Avoidance. The rationale for this is that the reception of three duplicate ACKs indicates that several segments have already been received, being it an strong indication that serious network congestion does not exist.

5.1.2 Retransmission timer

As previously mentioned, a retransmission timer is triggered whenever a data segment is sent and, if it expires without having received the ACK for that segment, it is retransmitted. The value of the TCP retransmission or RTO is computed based on RTT measurements performed by the emitter, that is, the time elapsed since the transmission of a given segment until the corresponding ACK is received. The minimum value of the obtained RTT samples (RTSs) is determined by the granularity of the clock used by the sender, with typical values of several milliseconds [Psaras and Tsaoussidis, 2007]. In this way, RTO values are adapted to the particular attributes of the TCP connection such as available bandwidth, delay, etc. The complete algorithm is described in [Paxson et al., 2000] and here we provide a summary of it.

When the first segment of the connection is sent, as there are no RTT samples, the RTO is initially set to an arbitrary value, typically of 3 s. Once a RTT measure has been made, the RTO is derived as follows. Let RTS_i be the ith RTT sample measured by the emitter. Then, the value of the retransmission timer RTO_i is computed as in (5.2), being *Smoothed Round-Trip Time (SRTT)* an estimation of the average RTT and *Round-Trip Time Variation (RTTVAR)* an estimation of the average deviation as in (5.4). As suggested in [Jacobson, 1988], (5.4) is usually computed by setting $\alpha = \frac{1}{8}$ and $\beta = \frac{1}{4}$.

$$RTO_i = SRTT_i + 4\ RTTVAR_i \tag{5.2}$$

$$\begin{aligned} RTTVAR_i &= (1-\beta)\ RTTVAR_{i-1} + \beta\ |SRTT_{i-1} - RTS_i| \qquad (5.3) \\ SRTT_i &= (1-\alpha)\ SRTT_{i-1} + \alpha\ RTS_i \end{aligned}$$

For the first sample RTS_0, the values $RTTVAR_0$ and $SRTT_0$ are computed as in (5.5).

$$\begin{aligned} RTTVAR_0 &= \frac{RTS_0}{2} \\ SRTT_0 &= RTS_0 \end{aligned} \tag{5.4}$$

The value RTO_i computed as explained above is then used to set the retransmission timer value for all segments transmitted until a new RTT is measured. However, there is an exception to this rule: if the retransmission timer expires for a given segment, a backoff algorithm is applied and the new RTO value is computed as in (5.5).

$$RTO_{i+1} = 2\ RTO_i \tag{5.5}$$

With each consecutive unsuccessful attempt, the RTO is doubled until a maximum value RTO$_{max}$ which depends on the implementation.

On the other hand, it is common to set a minimum RTO value for efficiency reasons. In the original standard [Paxson et al., 2000], a conservative approach was adopted by setting a minimum value of 1s, mainly due to the coarse clock granularity used by most Operative Systems (OSs) at that time (typically of 500 ms). However, as modern systems use finer clocks, smaller values such as 200 ms in Linux TCP or 400 ms in Solaris TCP are used nowadays [Psaras and Tsaoussidis, 2007].

5.1.3 TCP in wireless networks

Besides the above mentioned TCP Tahoe and TCP Reno, many other variants have been proposed in order to improve its performance, e.g., TCP NewReno [Floyd and Henderson, 1999] or TCP Sack [Mathis et al., 1996]. These variants were designed for wired networks where most packet losses are due to network congestion. Therefore, their congestion control mechanisms consider packet losses as signals of congestion, and react to them by reducing TCP transmission rate. However, several studies [Bakshi et al., 1997, Tian et al., 2005] have shown that these variants are inadequate for transmissions over wireless links. In wireless networks, the major sources of packet losses are changes in environmental conditions, terrestrial obstructions, reflections and disconnections due to user's mobility. Besides, it is well known that the TCP protocol is also especially sensitive to high variations of delay and bandwidth [Mascolo et al., 2001, Wei et al., 2006], which frequently occur in wireless networks due to changes in the path or in channel conditions.

In this regard, several proposals have been appeared in the literature to cope with these problems. Some approaches require the modification of the basic operation of the protocol, such as TCP New Jersey [Xu et al., 2004] or TCP Eifel [Ludwig and Katz, 2000]. Based on statistical analysis of the data sent through the connection, they are able to distinguish between losses due to congestion and wireless losses, and react to them accordingly. Other examples are Freeze-TCP [Goff et al., 2000] , which also makes use of cross-layer mechanisms so as to provide information to the transport layer about physical disconnections. Or TCP Westwood, which deals with bandwidth changes in the channel in order to avoid packet loss. On the other hand, other variants, such as Snoop TCP [Balakrishnan et al., 1997] or I-TCP [Bakre and Badrinath, 1994], rely on the existence of an intermediate entity, i.e. a proxy, between the TCP sender and receiver, which is responsible for data retransmission when errors at the link layer are detected. Although such proxy-based proposals outperform the traditional TCP protocol, they break the end-to-end semantics of the protocol. Besides, they require state maintenance at the intermediate entity in order to retransmit lost data.

5.2 The Lion attack

5.2.1 Target and motivation

The Lion attack is a cross-layer attack performed at the link-layer and targeted to the transport layer, aiming to degrade the throughput of TCP connections established in a CRN. It is based on performing a PUE attack in order to force a frequency handoff in the CRN which, as explained in Section 2.1.1, leads to the interruption of all communications in the network for a given period of time. This interruption can have a harmful impact over the throughput of TCP connections due to the interaction with its congestion control mechanisms, already described in Section 5.1.

Although frequency handoffs could also be forced by means of jamming, there are fundamental differences which may motivate an attacker to perform specifically a PUE and not simply jam the channel. First, a CRN is required to perform a frequency handoff upon detection of a primary transmission, even if the remaining channels offer worse transmission conditions. If the channel is jammed, the victim CRN may just perform the handoff if the overall transmission conditions are below a certain threshold and a better channel is available. Moreover, note that the cost of a PUE attack comes down to transmit a signal similar to a primary signal, e.g., TV or wireless microphones signals, or replay a real one. On the other hand, with the same effort or amount of resources the scope of a PUE attack can be much larger. Even if the fake primary transmission is only detected in a small area of the CRN, it can force a frequency handoff and affect the whole CRN. Contrary to it, the fact of degrading the communication channel only in a small area should not be enough to force the CRN to perform such handoff. Because of this, the Lion attack is more cost-effective in reducing TCP throughput than a simple jamming attack.

Furthermore, if the attacker knows or can guess some of the TCP connection parameters, the Lion attack can lead to a DoS. This can be achieved, as we will see in the next section, by emulating primary transmissions at specific instants of time, which can be easily predicted by the attacker.

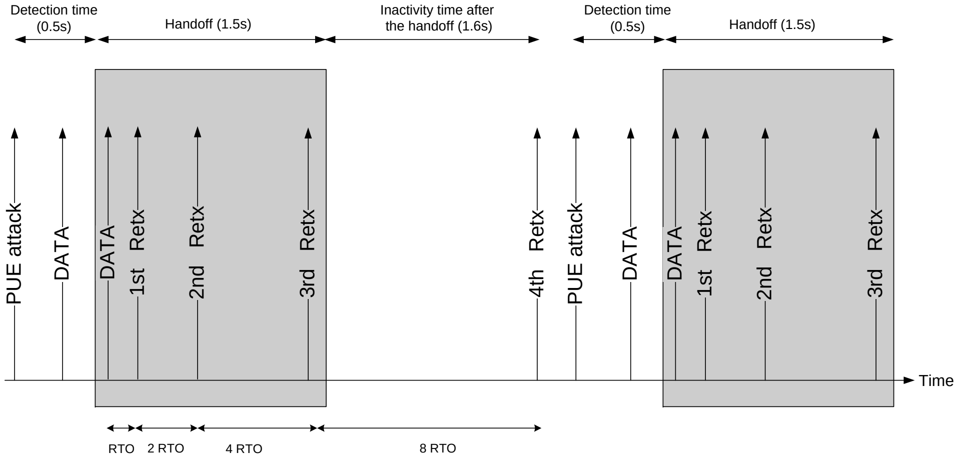


Figure 5.2: Lion attack

5.2.2 Attack insights

As mentioned in Section 5.1.3, the TCP protocol is especially sensitive to high variations of delay and bandwidth and therefore, the interruption of the transmission due to a frequency handoff can lead to a very poor performance. As the transport layer is not aware of the interruption, the TCP sender keeps sending data which is queued for transmission at lower layers. Thus, outstanding TCP segments can be delayed or even lost if the queue overflows during the process of spectrum handoff, triggering the TCP congestion control mechanisms.

If the retransmission timer for a given segment expires, i.e., a RTO takes place and no ACK has been received, the segment is considered to be lost and it is retransmitted. Besides, the cwnd is reduced to one segment thereby reducing the TCP throughput [Jacobson, 1988]. The expiration of the retransmission timer can be due to the lost of a segment but also to a sudden increase in the RTT. The latter may happen, for example, if there is a route change or when a spectrum handoff takes place in a CRN. Because spectrum handoffs can last for a few seconds, it is very likely that the RTO of the subsequent retransmitted segments expire too. With each unsuccessful attempt, the retransmission timer backs off, i.e., doubles its value. Besides, after a retransmission takes place, the TCP sender is not able to transmit new data because the transmission window is equal to only one segment. This implies that the TCP sender may remain inactive for a long time, even after the frequency handoff has finished.

Figure 5.2 depicts the effect of the attack considering an initial RTO of 200ms. A PUE attack is performed and the CRN detects the presence of a potential primary user after t_Ds. Then, the CRN performs a frequency handoff with a duration of $t_H = 1.5$ s. During the handoff, as the channel is not available, the data sent by the TCP sender is not acknowledged leading to the expiration of the retransmission timers. A retransmission attempt is performed 200ms after the first transmission of the segment and, since the handoff has not finished, it does not success. As a consequence, the TCP sender backs off doubling its retransmission

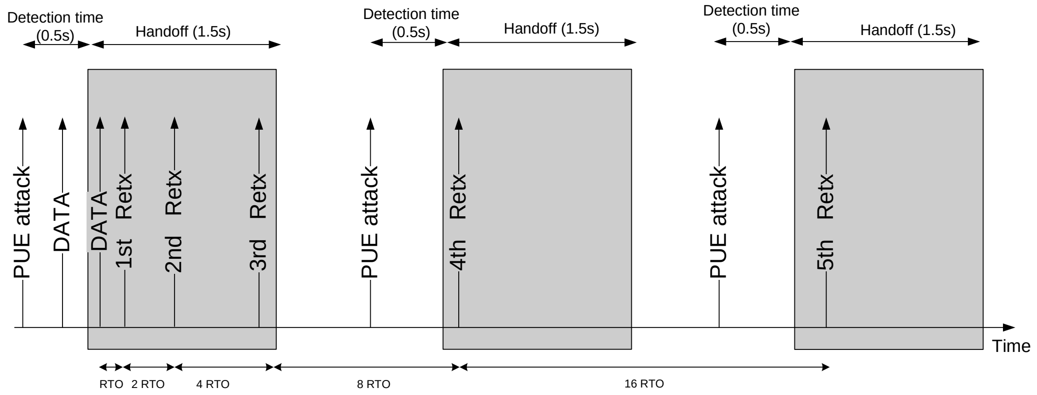


Figure 5.3: Smart Lion attack

timer and tries to retransmit the segment after 2·RTO= 400ms. All retransmissions matching a handoff interval will fail, triggering again the back off mechanism. In this example, the fourth retransmission attempt finally succeeds but the TCP sender has remained inactive for 15·RTO= 3s.

An smart version of the attack could be implemented based on the knowledge of the value of the retransmission timer of the TCP connection. In typical CRNs such as WRAN 802.22 networks [Cordeiro et al., 2006], the RTT value for in-network communications is around some hundreds of microseconds. Although the value of the retransmission timer is variable and depends on the RTT estimations, most implementations round off its RTO to a minimum value of typically 100ms or 200ms, much higher than the real RTT. This fact will lead the TCP sender to make use of a fixed value for the RTO, which will be doubled for each unsuccessful attempt. The attacker can take advantage of this information to force handoffs at the specific instants in which retransmission attempts are performed, therefore completely starving the TCP source, as shown in Figure 5.3.

5.3 Analytical Model

As explained in Section 5.2, a Lion attack can degrade the throughput of a TCP connection leading in some situations to the starvation of the TCP source. Because of this, there is a need for evaluating its impact on the performance of TCP, either via simulation or by means of analytical models. Although the latter generally make significant assumptions, they are faster in providing results with enough accuracy.

In this section we derive an analytical expression both for the average inactivity time and the percentage of inactivity of a TCP source due to the attack. It is important to remark that the presented model is just an approximation that is neglecting many marginal contributions. Its accuracy is nevertheless proved by comparing its results with those obtained by means of simulation in Section 5.4.

5.3.1 Mathematical background

Let S_k as in expression (5.6) be the sum of $k \in \mathbb{N}$ independent and i.i.d. random variables X_i, $i \in [1, k] \subseteq \mathbb{N}$, with probability density function (pdf) as in (5.7) and cumulative distribution function (cdf) as in (5.8).

$$S_k = X_1 + X_2 + \cdots + X_k = \sum_{i=1}^{k} X_i \tag{5.6}$$

$$f_{S_k}(t) = \left(f_{X_1} * f_{X_2} * \cdots * f_{X_k}\right)(t) \tag{5.7}$$

$$F_{S_k}(t) = \int f_{S_k}(t)dt \tag{5.8}$$

Lemma 5.3.1. *Given S_k as in (5.6), the probability of only and no more than $k \in \mathbb{N}$ events occurring within the interval $(t, t+\tau]$, $t \geq 0, \tau > 0 \in \mathbb{R}$ is:*

$$Pr(k \text{ events in } (t, t+\tau]) = F_{S_k}(\tau) - F_{S_{k+1}}(\tau)$$

Proof. Let us denote by $A = \{S_{k+1} : S_{k+1} \geq \tau\}$, $B = \{S_k : S_k \leq \tau\}$ and $C = \{S_k : S_k > \tau\}$. The probability of only and no more than $k \in \mathbb{N}$ events occurring within the interval $(t, t+\tau]$ can be expressed as the probability of $A \cap B$.

As $A = (A \cap B) \cup (A \cap C)$, being

$$Pr(A) = Pr(S_{k+1} \geq \tau) = 1 - F_{S_{k+1}}(\tau)$$
$$Pr(A \cap C) = Pr(C) = Pr(S_k > \tau) = 1 - F_{S_k}(\tau)$$

then

$$Pr(A \cap B) = Pr(A) - Pr(A \cap C) = F_{S_k}(\tau) - F_{S_{k+1}}(\tau)$$

□

5.3.2 Assumptions

In order to develop the model, the following assumptions have been adopted:

- A malicious user performs several attacks, each one leading to a frequency handoff.
- The duration of a handoff, which we denote by t_H, is fixed.
- The time needed in order to start a handoff after the CRN detects the presence of a primary user, i.e., the channel detection time, is fixed with value t_D.
- The time elapsed since the end of a frequency handoff until the attacker performs the next attack is modeled by a random variable X_i. Accordingly, we define $X'_i = X_i + t_D + t_H$ as a random variable that represents the time elapsed since the end of a handoff until

the end of the next one. As a result, we can define S'_k as a random variable being the sum of $k \in \mathbb{N}$ i.i.d. random variables X'_i as in (5.9), with pdf and cdf as in (5.10) and (5.11), being S_k the sum of $k \in \mathbb{N}$ i.i.d. random variables X_i as in (5.6).

$$\begin{aligned} S'_k &= \sum_{i=1}^{k} X'_i && (5.9) \\ f_{S'_k} &= f_{X'_1} * f_{X'_2} * ... * f_{X'_k} \\ &= f_{X_1} * f_{X_2} * ... * f_{X_k} * \delta(t - k(t_D + t_H)) \\ &= f_{S_k}(t - k(t_D + t_H)) && (5.10) \\ F_{S'_k}(t) &= F_{S_k}(t - k \cdot (t_H + t_D)) && (5.11) \end{aligned}$$

- The RTT of the TCP connection is always smaller than the minimum RTO of the TCP source RTO_{min}. As explained in Section 5.2, this can be assumed in CRNs such as 802.22 networks. We have also considered that, when the first handoff takes place, the value of the retransmission timer is RTO_{min}.

- With each unsuccessful retransmission attempt, the RTO value is doubled until a maximum value RTO_{max}, which is equal to RTO_{min} multiplied by a power of 2. As a result, the value of the RTO for the i_{th} retransmission can be expressed as in (5.12) and the set of possible retransmission instants t_i defined as in (5.13).

$$\begin{aligned} RTO_i &= \begin{cases} 2^{i-1} \cdot RTO_{min} & \text{if } i \leq i_{max} \\ RTO_{max} & \text{if } i > i_{max} \end{cases} && (5.12) \\ i_{max} &= \log_2 RTO_{max} + 1 \\ RTO_{max} &= 2^{i_{max}-1} \cdot RTO_{min} \end{aligned}$$

$$\begin{aligned} t_i &= \begin{cases} RTO_{min} & \text{if } i = 1 \\ t_{i-1} + RTO_i & \text{if } i > 1 \end{cases} \\ &= \begin{cases} (2^i - 1) \cdot RTO_{min} & \text{if } i \leq i_{max} \\ (i - i_{max} + 2) \cdot RTO_{max} - RTO_{min} & \text{if } i > i_{max} \end{cases} && (5.13) \end{aligned}$$

- As shown in Figure 5.4, we assume that it always takes place at least one handoff (handoff 0). Considering that the first segment loss takes place at the beginning of the handoff 0, the retransmissions attempts being performed at $t_i < t_H$ will fall in the handoff interval and therefore will fail. This implies that $Pr(t = t_i) = 0$. For the sake of clarity, we define a new time axis $t' = t - t_H$ and thus we redefine the retransmission instants as $t'_l = t_i - t_H$, being $t'_1 = t_s - t_H$, with s the index of the first t_i satisfying the condition $t_i > t_H$. As a result, $l = i - s + 1$ for $i \geq s$.

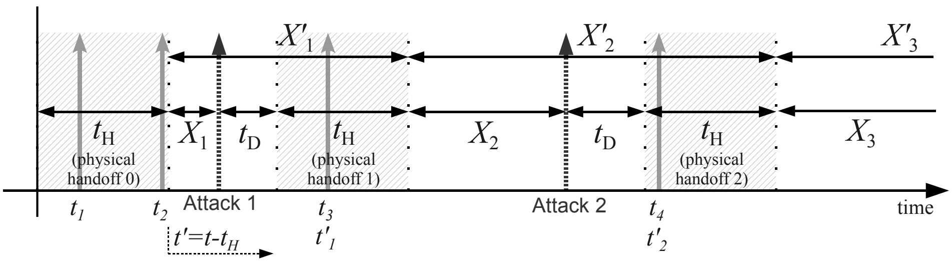


Figure 5.4: Analytical model for the Lion attack

5.3.3 Probability of k handoffs in interval $(t', t' + \tau]$

The probability $p_k(\tau)$ that k handoffs occur in the interval $(t', t' + \tau]$ is the probability of k events of the random variable X'_i in the interval $(t', t' + \tau + t_H]$ (see Figure 5.4). Therefore, from Lemma 5.3.1, $p_k(\tau)$ can be expressed as in (5.14).

$$p_k(\tau) \quad = \quad \begin{cases} 1 - F_{S'_1}(\tau + t_H) & \text{if } k = 0 \\ F_{S'_k}(\tau + t_H) - F_{S'_{k+1}}(\tau + t_H) & \text{if } k > 0 \end{cases} \tag{5.14}$$

5.3.4 Probability that a given instant t' coincides with the k_{th} frequency handoff

Let $h_k(t')$ be the probability function that a given instant t' coincides with the kth frequency handoff, given that k handoffs have occurred. An expression for $h_k(t')$ can be easily derived from Figure 5.4 as in (5.15).

$$\begin{aligned} h_k(t')|_{k>0} &= Pr\left(S'_k - t_H \leq t' \leq S'_k\right) \\ &= Pr\left(t' \leq S'_k \leq t' + t_H\right) \\ &= F_{S'_k}(t' + t_H) - F_{S'_k}(t') \end{aligned} \tag{5.15}$$

5.3.5 Probability distribution of the inactivity time

Let T be a discrete variable representing the inactivity time of a TCP source, i.e., the time elapsed since the beginning of a frequency handoff until the TCP source successfully transmits a segment. As explained in Section 5.3, the possible values of T, which we denote as t_i, can be obtained as the sum of all the RTO events taking place before a retransmission succeeds as in (5.13).

The probability $P_r(T = t_i)$ is equal to the probability that the instant of time $t = t_i$ does not fall in a handoff interval, given that the previous instants $t = t_j$ with $j = 1..i - 1$ have fallen in a handoff interval. For example, the inactivity time will be $T = 15 \cdot RTO_{min}$ whenever the retransmissions performed at instants $t_1 = RTO_{min}$, $t_2 = 3 \cdot RTO_{min}$ and $t_3 = 7 \cdot RTO_{min}$ fail, but the next attempt at $t_4 = 15 \cdot RTO_{min}$ succeeds.

Then, the probability $Pr(T = t_i)$ can be computed as in (5.16), with k_{max} the maximum number of handoffs which can take place during the interval $[0, t'_l]$ as in (5.17), and $k_{min} = l' - 1$ the minimum number of handoffs that must take place during the interval $[0, t'_l]$ in order to have an inactivity time of t_i, i.e., the number of retransmission attempts that fail at $t'_{l-1}, t'_{l-2}, ..., t'_1$ before the next one succeeds at instant t'_l.

$$Pr(T = t_i = t'_l + t_H) = \begin{cases} 1 - F_{S'_1}(t'_l) & \text{if } l = 1 \\ \sum_{k=k_{min}}^{k_{max}(t'_l)} p_k(t'_l) * \zeta(1,1,l,k) & \text{if } l > 1 \end{cases} \tag{5.16}$$

$$k_{max}(t') = \left\lceil \frac{t' - t_D}{t_H + t_D} \right\rceil \tag{5.17}$$

$$\zeta(l, j, l_{max}, k) = \begin{cases} \sum_{m=j}^{m_{max}} \big(h_m(t'_l) \cdot \zeta(l+1, m+1, l_{max}, k)\big) & \text{if } l < l_{max} \\ F_{S'_k}(t'_l) & \text{if } l = l_{max} \text{ and } j \leq k \\ 1 & \text{if } l = l_{max} \text{ and } j > k \end{cases} \tag{5.18}$$

$$m_{max} = \begin{cases} k - (l_{max} - l - 1) & \text{if } k - (l_{max} - l - 1) < k_{max}(t'_l) \\ k_{max}(t'_l) & \text{otherwise} \end{cases} \tag{5.19}$$

In (5.18), k represents the total number of handoffs that take place during the period $(t_H, t'_l + t_H)$; $j - 1$ the number of handoffs already performed until instant t'_{l-1}; $l_{max} - l - 1$, the number of handoffs which must take place after t'_l and coincide, each one of them, with instants t'_{l+1}, t'_{l+2}...until $t'_{l_{max}}$; and $m_{max} - j + 1$ the maximum number of handoffs that can take place until instant t'_l.

As an example, let's suppose that we want to compute $Pr(T = t_i = 6.2s)$ for a given connection with $RTO_{min} = 0.2$ s and $t_H = 1.5s$. The set of instants t_i to consider are $t_1 = 0.2s, t_2 = 0.6s, t_3 = 1.4s, t_4 = 3s, t_5 = 6.2$. Assuming that the first handoff is performed at $t = 0$, the first retransmission attempt will take place at t=0.2s. This retransmission will fail, since it will fall in the first handoff interval. The same will happen for the next retransmissions attempts performed at $t_2 = 0.6$s and $t_3 = 1.4$s, since the condition $t_i < 1.5s$ is met. A new retransmission attempt will take place at $t = 3$s, once the first handoff has ended, but in order to have an inactivity time of $T = t_i = 6.2$s, this retransmission should fail too. Otherwise, the inactivity time would be $T = t_4 = 3$s.

Since the first instant satisfying $t_i > t_H$ is $t_i = t_4$, now we can define $t'_1 = t_4 = 3s$ and $t'_2 = t_5 = 6.2s$, since $Pr(T = t_i) = 0$ for the previous instants. Then,

$$Pr(T = t_i = 6.2s = t'_2) = \sum_{k=k_{min}}^{k_{max}} p_k(t'_l) * \zeta(1,1,l,k) = \sum_{k=1}^{3} p_k(t'_2) * \zeta(1,1,l,k) =$$
$$p_1(t'_2) * \zeta(1,1,2,1) + p_2(t'_2) * \zeta(1,1,2,2) + p_3(t'_2) * \zeta(1,1,2,3)$$

with $k_{min} = 1$, since at least one handoff must take place at $t'_1 = 3$s. On the other hand, the maximum number of handoffs that can take place during an interval of 6.2s is $k_{max} = 3$, which can be obtained from (5.17).

If there is only one handoff during the interval (t_H, t'_i), it must coincide with $t'_1 = 3$s and therefore:

$$\zeta(1,1,2,1) = \sum_{m=1}^{1} h_m(t'_1) \cdot \zeta(2,2,2,1) = h_1(t'_1)$$

If there are two handoffs during the interval (t_H, t'_i), one of them must coincide with $t'_1 = 3$s and the second must not coincide with $t'_2 = 6.2$s. Otherwise the time of inactivity would be longer than $t'_2 = 6.2$s. Then:

$$\zeta(1,1,2,2) = \sum_{m=1}^{1} h_m(t'_1) \cdot \zeta(2,2,2,2) = h_1(t'_1) * F_{S'_2}$$

Finally, if there are three handoffs during the interval (t_H, t'_i), at least one of them must coincide with $t'_1 = 3$s and the last one must not coincide with $t'_2 = 6.2$s. Accordingly:

$$\zeta(1,1,2,3) = \sum_{m=1}^{1} h_m(t'_1) \cdot \zeta(2,2,2,3) = h_1(t'_1) * F_{S'_3}$$

5.3.6 Average inactivity time of a TCP source after an attack

Since T is a discrete random variable with a set of possible values t_i defined as in (5.13) and with probabilities $Pr(T = t_i)$ as in (5.16), the expected average time of the TCP source inactivity $\overline{T}$ after receiving an attack can be obtained as in (5.20).

$$\overline{T} = \sum_{i=1}^{\infty} t_i \cdot Pr(T = t_i) \tag{5.20}$$

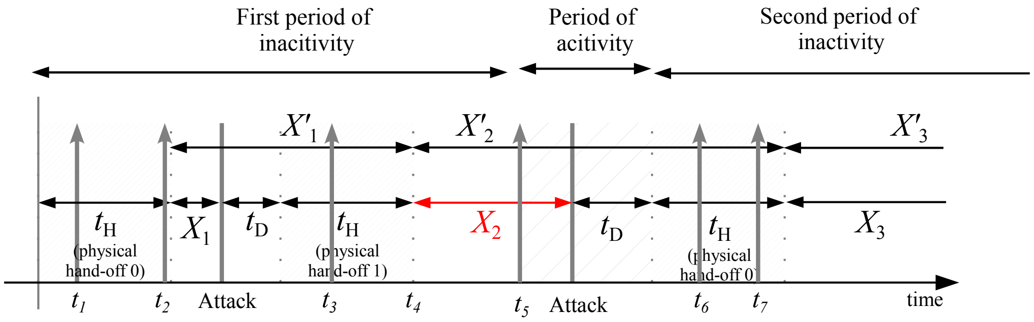


Figure 5.5: Activity time after a Lion attack

5.3.7 Percentages of TCP activity and inactivity due to an attack

We can assume an inactivity percentage $U_{inactivity}$ as in (5.21), or, the other way round, a percentage of activity $U_{activity}$ as in (5.22). These parameters allow to measure the degradation of TCP throughput due to an attack.

$$U_{inactivity}(\%) = \frac{\overline{T}}{\overline{T}+\overline{A}} \times 100 \tag{5.21}$$

$$U_{activity}(\%) = \frac{\overline{A}}{\overline{T}+\overline{A}} \times 100 \tag{5.22}$$

$\overline{T}$ is the average inactivity time of the TCP due to the attack and can be obtained as in (5.20). The average activity time $\overline{A}$ is the average time elapsed since the end of a frequency handoff until the next one starts and can be computed as in (5.23).

$$\overline{A} = E[X_i + t_D] = E[X_i] + t_D \tag{5.23}$$

However, the value given by (5.23) is only a rough approximation of the time of activity. For the sake of clarity, consider the case depicted in Figure 5.5. A first handoff is performed at $t = 0$ and leads to a inactivity time of t_5 s, since the retransmissions performed at instants t_1,t_2,t_3 match the handoff interval and therefore fail. When finally a retransmission succeeds at instant t_5, it starts a period of activity which will last until the beginning of the next handoff, i.e., at instant $t_4 + X_2 + t_D$, with X_2 the interval of time represented by a red arrow. From this reasoning, the time of inactivity should be computed as $E[X] + t_D - (E[t_5] - E[t_4])$. The values of t_4 and t_5 depend on the number of handoffs that have occurred until that moment, and on the specific amount of time elapsed between consecutive handoffs. As a consequence, we cannot derive an exact expression for the time of activity. The time of activity given by (5.23) is larger than the real one because we are considering that the period of activity ends at instant $t_5 + X_2 + t_D$. Therefore, we can only provide an upper bound for such time of activity, and the minimum percentage of inactivity as in (5.25).

$$\begin{aligned} \overline{A} &\leq E[X_i] + t_D & (5.24) \\ U_{inactivity}(\%) &\geq \frac{\overline{T}}{\overline{T} + \overline{A}} \times 100 \end{aligned}$$

5.4 Impact of the Lion attack

With the purpose of evaluating the impact of the attack on the performance of a TCP connection, we have conducted a set of simulations with the ns-2 simulator [USC/ISI et al., 2007]. We have assumed that the network cannot detect PUE attacks and that the control data channel is secured. The rationale behind this is that, if the PUE attack is detected, the CRN will remain in the same channel and therefore no frequency handoff will take place. Furthermore, if the victim network uses an insecure control channel, the attacker can easily obtain the next operational channel and perform a DoS. In such case, simulation results are of little value since the TCP throughput would be completely reduced to zero.

Besides, the model presented in Section 5.3 has been programmed in Matlab [MATLAB, 2009] and it has been validated by comparing its results with the ones provided by the simulations.

5.4.1 Simulation scenario

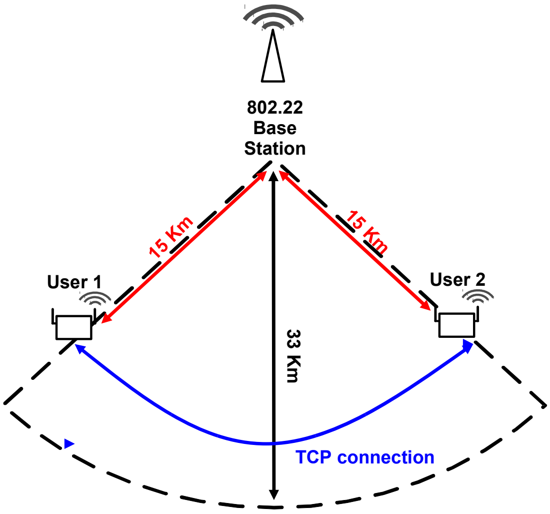


Figure 5.6: Simulation environment

Figure 5.6 depicts the simulated environment, consisting in a 802.22 network which access to unused TV spectrum bands in an opportunistic manner. We assume that a 6 MHz TV channel is available for the operation of the network and that the spectral efficiency is of 3 bits/(s/Hz) (802.22 specifications define spectral efficiencies ranging from 0.5 bit/(s/Hz) to 5 bit/(s/Hz)), which leads to a channel transmission capacity of 18 Mbps. This channel is used to establish a TCP connection between two secondary users.

The 802.22 standard defines a signal coverage of up to 33 Km for 4 W EIRP. Thus, we have considered an average distance between both secondary users and the BS of 15 Km and a propagation delay of 50μs, obtained by considering a propagation velocity equal to the speed of light. Both the process delay at the BS and the Bit Error Rate (BER) have been neglected in order to reflect only the effects of the handoffs on TCP throughput.

We assume that the attacker must sense the medium in order to detect the next channel to be used by the CRN after the handoff. If we consider a 45% of the TV channels in use, there are 36 channels for the CRN operation out of 67 TV channels available in the UHF and VHF bands. Primary transmissions should not be interfered, so at least there must be 2 empty channels between every pair of TV channels in use [Cordeiro et al., 2006]. This fact reduces the amount of available channels for the operation of the CRN to 12. Considering a channel sensing time of 46.95ms [Chouinard et al., 2006] for detecting the occupation of a given channel, it would take to the attacker an average time of $\frac{12+1}{2} \cdot 46.95\text{ms} = 305.175\text{ms}$ to discover the new channel of operation. Note that, even if the attacker manages to discover the channel in such period of time, it still needs to tune its device with the proper parameters.

According to the previous reasoning, we have modeled the time elapsed since the end of a handoff until the next attack begins, as an exponential random variable with different mean values $\frac{1}{\lambda}$. Although to get more realistic results other random distributions could be more suited, we have selected an exponential distribution for ease of modeling. Notice that the sum of k of exponential random variables, that is the base of the analytical model, can be easily computed as a gamma distribution.

Once the attacker has discovered the channel of operation of the CRN, it performs the attack. Then, it takes t_D seconds to the CRN to detect the fake primary transmission and start the frequency handoff. The values for the detection time and the handoff duration are set to typical values of $t_D = 500\text{ms}$ and $t_H = 1.5\text{s}$ [Cordeiro et al., 2006], respectively.

The variant of TCP used in the simulations is TCP Reno. The sender is fed by a File Transfer Protocol (FTP) source and generates TCP segments of 1040 bytes, i.e., 1000 data bytes and 40 bytes for the TCP header. Taking into account that the RTT value for this scenario is much below 100 ms (5.26), we have considered a minimum value for the retransmission timer of $RTO_{min} = 200\text{ms}$. Furthermore, a maximum value of $RTO_{max} = 12.8\text{s}$ has been set, which is the default value in the simulator ns-2.

5.4.2 Results

Figure 5.7 depicts the instantaneous and the average throughput obtained for the TCP connection, when the average time between the end of a handoff and the beginning of the next attack is $\frac{1}{\lambda} = 305.175\text{ms}$. The red-shaded areas represent each handoff interval performed

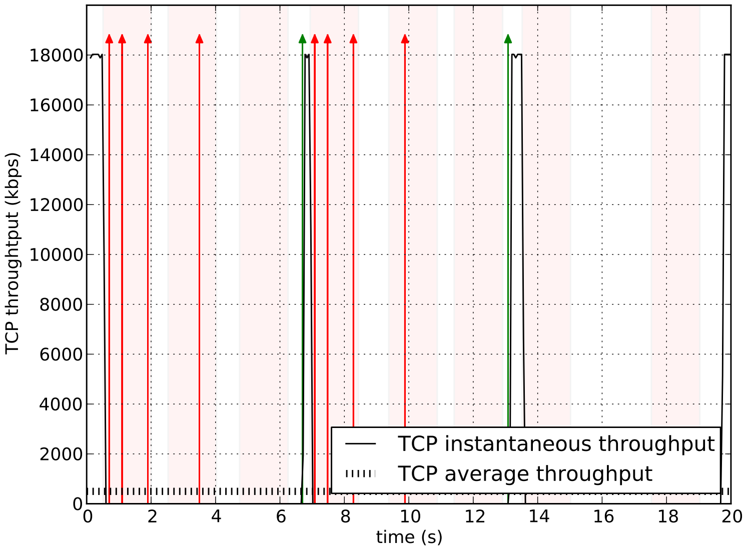


Figure 5.7: Effect of the Lion attack on TCP throughput

after an attack is received, in which the channel is not available. As it can be observed, due to the high frequency of attacks the TCP throughput is severely reduced, since most of the time the transmission is interrupted due to a handoff. The average throughput is of just 500kbps, a much lower value than the 18Mbps it should reach in absence of the attack. Besides, even in those periods of time in which the channel is available, the TCP sender remains inactive. The reason for such behavior is, as explained in Section 5.2.2, that during handoff intervals the sender continues transmitting data because is unaware of the disconnection. These data cannot be transmitted through the link and is buffered at lower layers, thus incurring high delays in the reception of the corresponding ACKs and/or packet loss, and leading to the expiration of the retransmission timers. This fact reduces TCP throughput because of two causes: 1) after a RTO takes place the congestion window is reduced to 1 segment; and 2) every time a segment is retransmitted, the retransmission timer is doubled (until it reaches a maximum value). The latter increases the inactivity time, since the TCP sender is not allowed to transmit any data until the next retransmission timer expires. The former almost does not affect our CRN, since the optimal window value for the connection is, as show in expression (5.26), just one segment. However, in networks with a larger time-bandwidth product, it would also contribute to the degradation of the throughput.

$$RTT = t_{tx} + 2t_{prop} \quad \approx \quad 641\mu s \tag{5.25}$$

$$W_{opt}(segments) = \frac{RTT}{t_{tx}} \quad \approx \quad 1.42 \tag{5.26}$$

As shown in expression (5.13), the time between consecutive retransmissions for a given segment is doubled with each unsuccessful attempt. Because of this, if a segment trans-

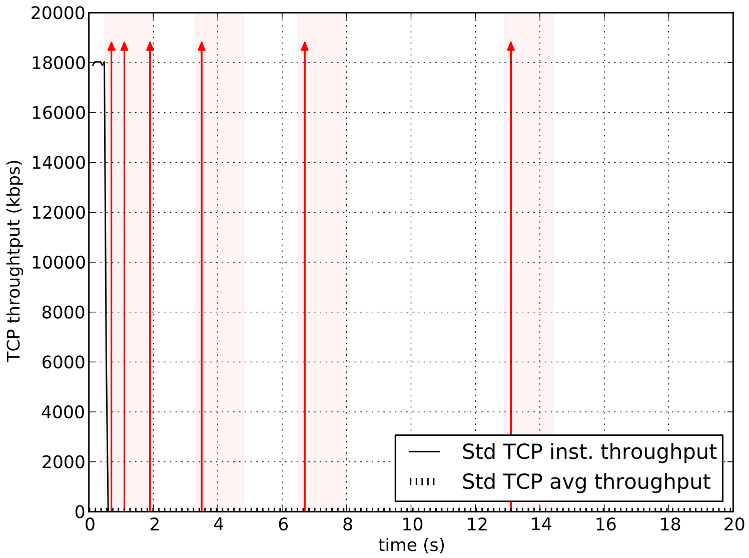


Figure 5.8: Effect of the smart Lion attack on TCP throughput

mission fails at $t = 0$, the corresponding retransmission attempts will take place at $t = [0.2s, 0.6s, 1.4s, 3s, ...]$. The retransmission attempts are represented in Figure 5.7 as red arrows, if the link is not available and therefore the retransmission fails, and as green arrows otherwise. The first handoff takes place at $t = 0.5$s with a duration of 1.5s. The first retransmission is performed 200ms after the beginning of the handoff, at $t =$ 0.5s+0.2ms= 0.7s, before the end of the handoff. The next take place at $t =$ 0.7s+0.4s= 1.1s and $t =$ 1.1s+0.8s= 1.9ms, as well within the period of handoff. At $t =$ 0.5s+1.5s= 2s the first handoff ends, but the TCP sender remains inactive (waiting for the expiration of the retransmission timer) until $t =$ 1.9s+1.600s= 3.5s. By that time, the attacker has forced another handoff and therefore all retransmissions fail again until $t =$ 3.5s+3.2s= 6.7s. In that instant, a retransmission finally coincides with a period of communication and therefore it succeeds. However, as it can be observed, the TCP connection has been inactive around 6.2 seconds.

If a smart Lion attack is performed, the attacker can detect the new operational channel through local sensing and predict the retransmission instants. In this way, it can force the handoffs to coincide with the retransmissions attempts leading to a null TCP throughput, as shown in Figure 5.8.

Table 5.1 reflects the time of activity $\overline{A}$, the time of inactivity $\overline{T}$ and the percentage of inactivity $U_{inactivity}$ of the TCP source when the attacker performs several attacks for different mean values $\frac{1}{\lambda}$. Both, the results extracted from the simulations and those derived from the analytical model are presented. For the analytical model, the maximum time of activity and the minimum percentage of inactivity are given, according to (5.25).

The results clearly reflect that most of the time the TCP sender is unable to transmit any

$\frac{1}{\lambda}$ (s)	Simulation			Analytical model		
	$\overline{A}$(s)	$\overline{T}$ (s)	$U_{inactivity}$(%)	$\overline{A}$(s)	$\overline{T}$ (s)	$U_{inactivity}$(%)
0.1	0.296	30.06	99.02	$\leq$0.6	30.63	$\geq$98.07
0.305	0.55	16.83	96.83	$\leq$0.805	16.64	$\geq$95.38
0.5	0.72	17.15	95.97	$\leq$1	17.46	$\geq$94.58
1	1.15	11.89	91.18	$\leq$1.5	11.63	$\geq$88.57
1.5	1.62	8.6	84.14	$\leq$2	8.37	$\geq$80.71
2	2.09	6.69	76.19	$\leq$2.5	6.48	$\geq$72.16
5	5.06	3.93	43.71	$\leq$5.5	3.76	$\geq$40.60

Table 5.1: Activity and inactivity time

data. In particular, for values of $\frac{1}{\lambda}$ below 1s the percentage of inactivity is above 90%. The time of inactivity is always much higher than the duration of the handoff $t_H = 1.5$s, due to the back off algorithm applied by the TCP sender. Obviously the more frequent the attacks, the longer periods of inactivity because the probability that a frequency handoff coincides with a retransmission attempt increases. If the attacker manages to rapidly guess the next channel to be used by the CRN and force a new handoff, i.e., in $\frac{1}{\lambda} = 0.1$s, the average time of inactivity rises to 30s.

Figure 5.9 shows the probability distribution of the time of inactivity for different values of $\frac{1}{\lambda}$. The probabilities for values of T above 51s are not presented in the histogram, since they are negligible compared to the rest, except for $\frac{1}{\lambda} = 0.1$s. On the other hand, the probability of $T = 3$s is zero for all values of $\frac{1}{\lambda}$ due to the duration of the handoff, i.e., $t_H = 1.5$s.

It can be observed that for $\frac{1}{\lambda}$ below 1s the most frequent time of inactivity is $T = 6.2$s. However, as the frequency between attacks is decreased, the probability of an inactivity time $T = 3$s is gradually increased until a value of 0.8163 for $\frac{1}{\lambda} = 5$s. From this results, it can be inferred that such probability tends to 1 as $\frac{1}{\lambda}$ grows. Also note that for $\frac{1}{\lambda} = 0.1$s, the most probable value is $T = 6.2$s but the average value is roughly $\overline{T} = 30$s. This is because around the 10% of the samples are above 100s and, as a consequence, the average time of inactivity is considerably increased. It is quite a reasonable result since the frequency of the attacks is extremely high.

On the other hand, by comparing the results obtained through the analytical model with the simulation results, it can be observed that the derived model can predict the time of inactivity due to the attack with considerable accuracy. The largest differences between both type of results can be found in the time of activity since, as previously mentioned, we have only provided an upper bound for such parameter. Despite it, it can be observed that as the values of $\frac{1}{\lambda}$ increases, the time of activity obtained in the simulations tends slowly to the value given by (5.25).

Note that this model can be valid for any probability distribution and, as a consequence, it can be used to analyze different attack patterns. As an example, it could be applied to

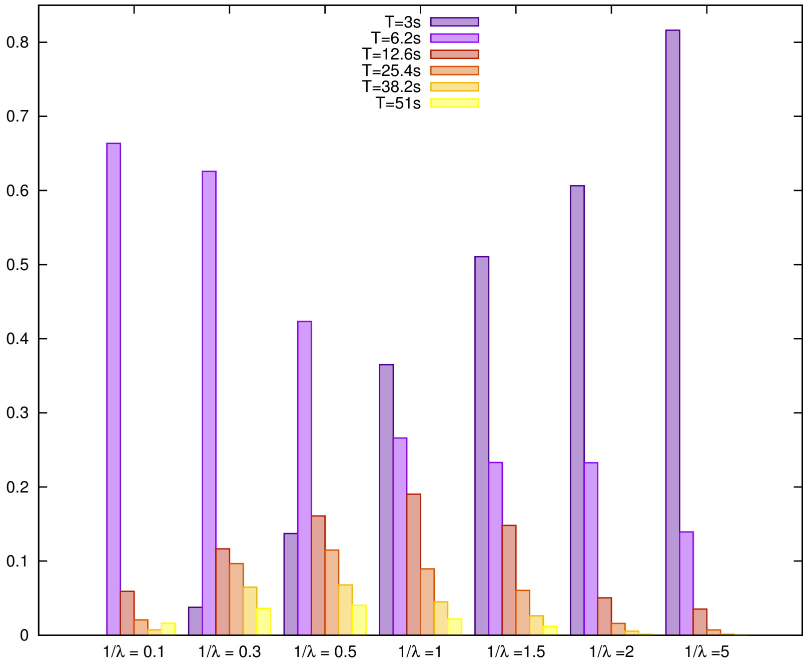


Figure 5.9: Probability distribution of T

analyze the impact on TCP of temporal disconnections due to other phenomena such as noise or user mobility by choosing the proper distribution.

5.5 Countermeasures for the Lion attack

As seen in Section 5.4, a Lion attack forces the CRN to perform a frequency handoff, incurring a substantial delay until transmission is resumed and degrading the TCP throughput. Because of this, CRNs should be provided with mechanisms in order to counteract this attack. In this section we propose the use of a TCP variant which can mitigate the effect of the attack. Moreover, we provide a set of guidelines for the design of mechanisms in order to detect and avoid it.

5.5.1 A handoff-tolerant TCP variant

As shown by the simulation results provided in Section 5.4, the TCP throughput is reduced under a Lion attack, not only because of the interruption of the communications during handoffs but also due to lack of information at the transport layer about physical disconnections. In order to overcome this problem, there is a need for making TCP aware of such interruptions and react to them accordingly.

As explained in Section 5.1.3, many TCP variants have been proposed in the literature to deal with typical TCP problems in wireless links, such as losses, drastic changes in routes or temporal lost of connectivity. Among them, Freeze-TCP [Goff et al., 2000] seems to be the most suited to deal with frequency handoffs. This TCP variant was designed to improve TCP performance in mobile environments where temporal disconnections occur frequently. In Freeze-TCP, the receiver is responsible for monitoring the signal strength to predict disconnections and advertising a zero-window to the sender before the disconnection takes place. Upon the reception of a zero window size, the sender enters the Zero Window Probe (ZWP) mode, in which it "freezes" its transmission parameters, i.e., congestion window, retransmission timers, etc. and it is not allowed to transmit any data. By means of this mechanism, it is possible to avoid potential losses and prevent the congestion window from dropping, because no retransmission timers expire during the handoff. When the connection is resumed, the receiver advertises a non-zero window which allows the sender to resume its transmission.

We propose to use a modified version of Freeze-TCP in which the TCP sender is responsible for freezing itself its own parameters without the need of being warned by the receiver, as it is the case of Freeze-TCP. Since in a CRN all members share information about the channel, the sender itself could predict the disconnection due to an incoming frequency handoff [León et al., 2009a].

Evaluation

In order to analyze the benefits introduced by this new TCP variant, we have conducted a set of simulations considering the same scenario as in Section 5.4. Figures 5.10 and 5.11 represent the effects of the Lion attack on TCP throughput when using both the proposed TCP implementation and the standard TCP Reno. Figure 5.10 depicts the basic attack while in Figure 5.11, the attacker performs handoffs matching the retransmission attempts of the TCP sender, i.e., it is performing a smart Lion attack.

As it can be seen, the TCP throughput is clearly higher when freezing TCP parameters than without freezing, since the TCP source only remains inactive during the handoffs and makes the most of the available transmission time. An interesting observation is that the throughput of the proposed TCP variant is higher when an smart Lion attack is performed than with the standard attack. This is due to the fact that freezing TCP parameters during the handoff avoids unnecessary retransmissions. As the attacker forces handoffs only at potential instants of retransmissions, which are each time more infrequent, the TCP sender is able to transmit for a longer period of time. This implies that, when the TCP source is freezing its parameters during a handoff, the knowledge of the attacker about the TCP connection is completely useless.

Table 5.2 reflects the inactivity and activity times of the modified TCP source, and the percentage of inactivity $U_{inactivity}$ obtained by means of simulation and the analytical model. When TCP parameters are freezed during the handoff, the back off algorithm is not triggered. As a consequence, the average time of activity can be computed as in (5.23) and the time of inactivity is always equal to the duration of the handoff $t_H = 1.5$s. Note that in this case, the analytical model and the simulations provide identical results. It can be observed that the

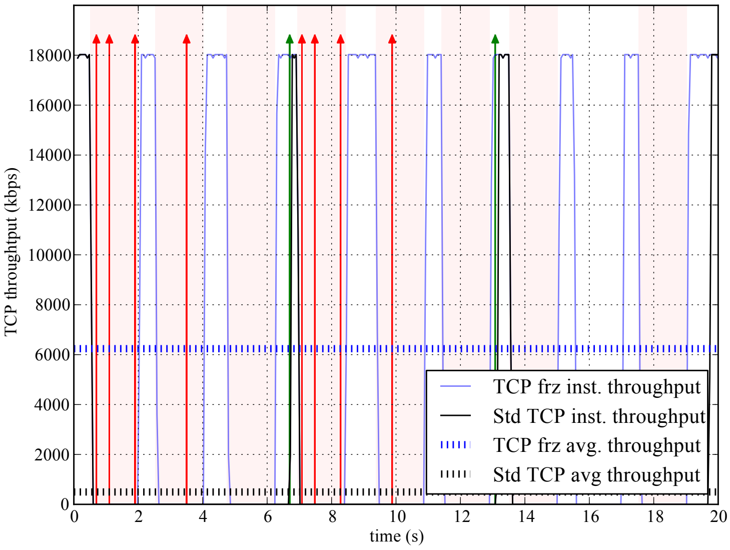


Figure 5.10: Effect of the Lion attack on TCP throughput freezing and non-freezing

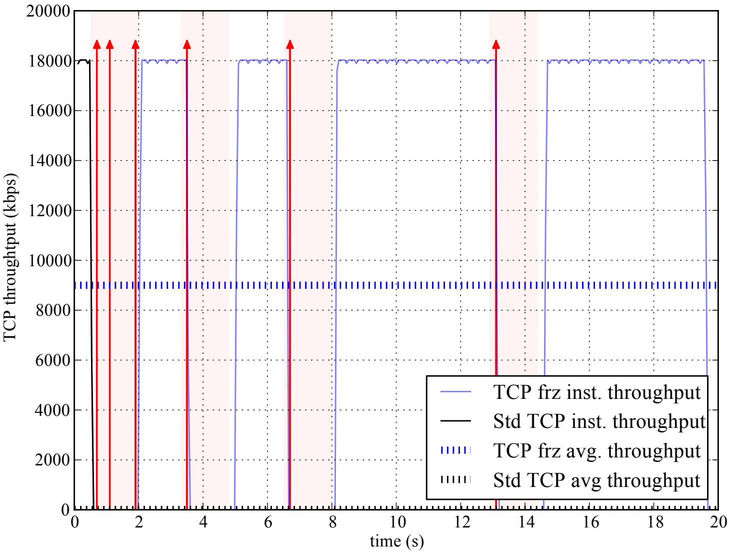


Figure 5.11: Effect of the smart Lion attack on TCP throughput freezing and non-freezing

proposed TCP variant clearly outperforms the standard protocol, especially for low values $\frac{1}{\lambda}$. As an example, for $\frac{1}{\lambda} = 0.305$s, the percentage of inactivity is reduced from roughly a 96% to a 65.21%.

$\frac{1}{\lambda}$ (ms)	Simulation			Model		
	$\overline{A}$(s)	$\overline{T}$ (s)	$U_{inactivity}$(%)	$\overline{A}$(s)	$\overline{T}$ (s)	$U_{inactivity}$(%)
0.1	0.602	1.5	71.36	0.6	1.5	71.42
0.305	0.8041	1.5	65.1	0.805	1.5	65.07
0.5	0.999	1.5	60.02	1	1.5	60
1	1.51	1.5	49.83	1.5	1.5	50
1.5	2.03	1.5	42.49	2	1.5	42.85
2	2.56	1.5	36.94	2.5	1.5	37.5
5	5.52	1.5	21.36	5.5	1.5	21.42

Table 5.2: Activity and inactivity time freezing parameters

As previously mentioned, even if the attacker is aware of the freezing mechanism used by TCP connections during the handoffs, it cannot take advantage of this information in order to improve the attack. The fact is that freezing TCP parameters limits the attacker to only degrade the TCP throughput, since it does not perform retransmissions attempts during the handoff. However, if the attacker manages to perform frequency handoffs whenever the CRN switches to a new channel, it can produce a permanent DoS attack. In order to avoid it, the CRN must prevent the attacker from rapidly detecting the next spectrum band to be used. Assuming that the attacker is also a CR device, it can predict the next frequency in the two following ways: 1) by sensing the medium; and 2) by eavesdropping the common control channel used in the CRN to exchange sensing information. Notice that the common control channel provides the attacker with a priori knowledge of the next operation channel, while sensing requires a given amount of time until the attacker discovers the new channel. Consequently, securing the control channel should be mandatory in any CRN network [León et al., 2010]. The IETF 802.22 WG has already considered such risk and the current standard [802.22 WG, 2011] defines a security sublayer providing features such as authentication, authorization, message integrity, and data encryption for data and control channels.

5.5.2 Further countermeasures

As shown in the previous section, neither the use of the proposed TCP variant nor securing the control data channel can avoid the attack but only mitigate its effect. Therefore, further countermeasures are required in order to efficiently detect and avoid the attack.

The smart Lion attack follows a well defined pattern and therefore it could be detected by means of a signature-based scheme, one of the detection methodologies described in Section 4.1.3. This mechanism should correlate cross-layer data, i.e., it should look for matches between retransmission attempts in a given TCP connection and the beginning of a frequency handoff, performed upon detection of a potential primary user. According to the degree of correlation, a probability of being under a Lion attack should be computed and compared with a predefined threshold. When crossing such threshold, an alert of attack should be generated. Note that this mechanism could be included in the system proposed in Chapter 4 in order to provide a second line of defense for CRNs.

As an example, let us consider a TCP connection with an initial retransmission timer of τ seconds. If a segment is lost due to a frequency handoff forced by an attack, it will be retransmitted after τ seconds. Since the TCP backoff algorithm doubles the retransmission timer with each unsuccessful attempt, next retransmission attempts will occur after $t = \tau, 3\tau, 7\tau, 15\tau, \ldots, (2^i - 1)\tau$ seconds. Considering an initial value for the retransmission timer of τ = 200ms and a handoff duration of t_H =1.5s, the retransmissions will be performed at instants $0.2, 0.6, 1.4, 3, 6.2, \ldots$ seconds. The first three retransmission attempts will obviously fail because they match the first handoff period, so they are not taken into account to increase the probability of an attack. At $t = 1.5s$, the handoff has ended and the CRN is operating in a new channel, so subsequent retransmissions should succeed. However, if a malicious user is performing a Lion attack, it may predict the instants of the next retransmission and force a new handoff, leading again to the failure of such retransmission attempt. The attacker may repeat this process each time the CRN performs a frequency handoff, completely starving the TCP source.

Consider a naïve implementation of the detection mechanism, in which we define a threshold equal to 4 retransmission failures matching the beginning of a handoff (excluding the first one). Also, consider that the probability of being under a Lion attack is increased by $\frac{1}{4} = 0.25$ with each new match, and that an alarm is generated when such probability is equal to 1. Then, upon the fourth retransmission failure at t = 3s, the module output would compute a probability of $\frac{100\%}{4}$ = 25%, a probability of a 50% with the fifth retransmission attempt at t = 6.2s , etc. Finally, with the sixth retransmission, we would get a probability equal to 1 of being under a Lion attack, and an alert for this attack would be reported.

The standard Lion attack is also based on forcing handoffs but in such case, there is no a clear relation between TCP retransmission attempts and the beginning of those handoffs. This implies that the above presented countermeasure would not be so effective in detecting the attack and therefore, an alternative mechanism should be applied. Given that the Lion attack relies on primary emulation transmissions, the most straightforward solution would be to use a PUE detection scheme. In chapter 6, we deal with this challenge and propose a method in order to detect PUE attacks.

5.6 Conclusions

The first step in order to secure CRNs is to identify new potential threats and evaluate their impact on the performance of such networks. In this chapter we have presented a new cross-layer attack to TCP in CRNs, which relies on emulating licensed transmissions in order to force frequency handoffs in the CRN. During a frequency handoff, the channel is not available for transmission and communications must be interrupted, leading to packet losses and high delays. Due to the unawareness of the transport layer about physical disconnections, such effects are miss-interpreted by TCP connections as signals of network congestion. This fact implies a dramatic reduction of TCP throughput and can even produce a DoS, provided that the attacker can predict the retransmission instants of TCP data. By performing a smart Lion attack, i.e., by forcing frequency handoffs at the specific instants in which retransmissions occur, the attack can lead a TCP sender to remain inactive for long intervals of time.

An analytical model has been derived in order to compute the average time of inactivity and the percentage of inactivity of a TCP sender due to the attack. Such model can be used for different attacker patterns regarding the probability distribution of the time between consecutive attacks.

The impact of the attack on TCP performance has been analyzed both by means of the analytical model and via simulation. The results show that a given connection can remain inactive for several seconds, even when the handoff has ended and the CRN is operating on a new channel. The percentage of inactivity increases with the frequency of the attacks, and can rise to 99% for the standard attack, implying that the TCP sender is almost starved. On the other hand, when a smart Lion attack is performed, the throughput of TCP dramatically decreases to zero.

With the aim of mitigating this attack, we have proposed a set of countermeasures. First, we have suggested some modifications to the TCP protocol in order to avoid the degradation of the throughput due to frequency handoffs. These modifications imply a cross-layer mechanism to exchange data between the physical/link and the transport layers. In this way, a TCP sender is able to freeze its transmission and congestion control parameters during frequency handoffs. This countermeasure has also been evaluated by means of simulation and it has been shown that freezing TCP parameters reduces the effect of the handoffs on the throughput of TCP. Besides, it prevents from a DoS when a smart Lion attack is performed.

A second countermeasure has been proposed in order to specifically deal with smart Lion attacks. It relies on a signature-based detection method, and it is intended to detect such attacks by correlating data from several layers, i.e., from the physical and the transport layers. Finally, we have stressed the need for a PUE detection method. Since the Lion attack can be performed by means of a primary transmission emulation, such detection methods could also be used in order to counteract both the standard and smart Lion attacks. This challenge will be addressed in the next chapter.

Chapter 6

Cooperative Detection of PUE attacks in CRNs

One of the most important challenges that CRNs must face is the effective detection of white spaces in the spectrum while ensuring incumbents protection. As explained in Section 1.1.3, several spectrum sensing mechanisms have been studied by the research community in order to achieve these goals, but their performance is limited due to the inherent characteristics of the radio environment. Besides, the spectrum sensing process can be disrupted by a malicious user performing a PUE attack.

Existing countermeasures for the PUE attack are usually based on energy measures of the received signal which are provided by members of the CRN. These measures are combined in a fusion center, typically the BS, in order to detect whether a given emitter is a legitimate primary user or not. However, these approaches can be easily overcome by an attacker by selecting an optimal position and properly adjusting its transmission power.

Location of the transmission source can be a valuable tool to detect such attacks whenever the position of true primary transmitters is known, as it is the case of TV towers in IEEE 802.22 networks. Therefore, in this chapter we present a cooperative location method to effectively deal with PUE attacks in such scenarios. The position of a potential attacker is estimated based on a set of time measures, which are derived from the feedback provided by members of the CRN. Besides, a heuristic approach is adopted to improve the accuracy of the estimation. The decision about the existence of a primary user or an attacker is then performed by comparing the position estimate with the known positions of true primary users. This method is related to the cooperative localization module and the PUE detection module of the security system proposed in Chapter 4, now gray-shaded in Figure 6.1. This chapter is structured as follows. Section 6.1 provides an overview of the main location techniques used in wireless networks and analyzes their suitability for CRNs. Section 6.2 is devoted to the description of the proposed location method, and its goodness is proved via simulation in Section 6.3. Finally, in Section 6.4 we provide the conclusions of this work.

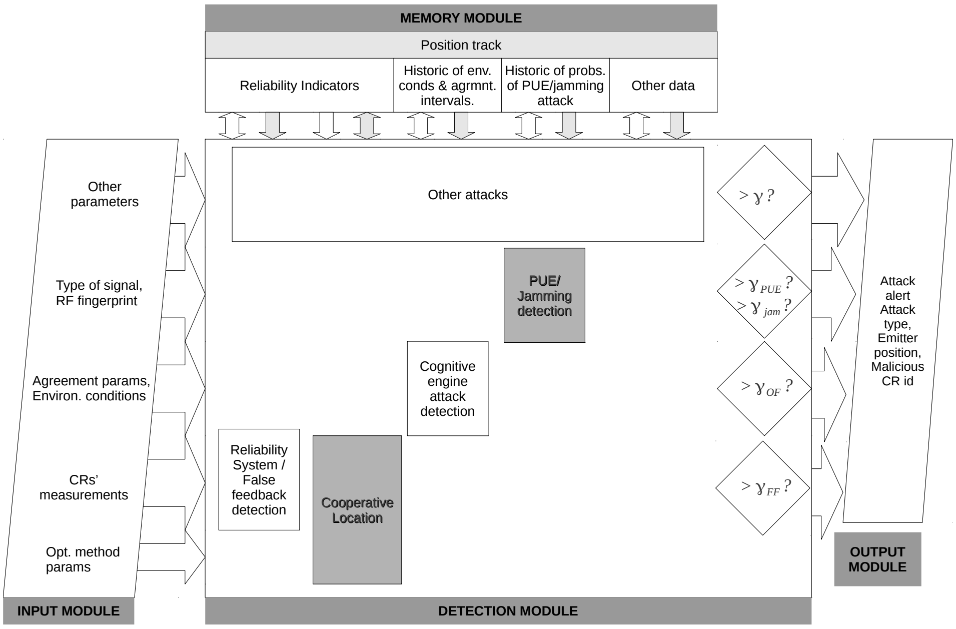


Figure 6.1: Cooperative Detection of PUE attacks modules

6.1 Location techniques

Physical location of RF transmission sources has been a hot topic for many years in wireless applications, especially for those related to military and safety purposes. Recently, its interest has increased due to its application to a wide variety of civil services such as E-911 telephone service, traffic routing, fleet management, etc.

Wireless location techniques fall into two main categories: mobile based and network based. In mobile-based location systems, a given mobile node computes its position by means of signals received from some BSs or from Global Positioning System (GPS) satellites. On the other hand, network-based technologies rely on measures of certain distance dependent parameters, performed at a set of reference nodes whose position is known, which we will refer to as anchor nodes. Distance or angle information is then used to estimate the position of the node, typically by applying a method such as triangulation, trilateration or multilateration. The selection of the method will depend on the type of measurement performed and the available number of observations.

Mobile based approaches require a node to report its own position and therefore, they are not suitable for locating an attacker but an actual legitimate node of the network. Because of this, from now on we will focus on network-based techniques.

6.1.1 Distance/angle estimation

The distance or angle between two nodes can be obtained by means of several methods, such as RSS, ToA, TDoA or AoA techniques. In the following section, we provide an overview of such methods and study the feasibility of their application to CRNs.

RSS

RSS-based techniques rely on the fact that signal strength varies inversely with a power of the distance $d^{-\alpha}$ in free space, where α is the path-loss exponent. Assuming that the transmission power of the source and the path loss model are known, it is possible to estimate the distance from the source to an anchor node. The path-loss exponent depends on the type of environment, i.e., urban, sub-urban, rural..., and typically takes values between 2 and 4 dB. According to this, the RSS at distance d is typically modeled as in (6.1),

$$P(d) = P_0 - 10\alpha \log(\frac{d}{d_0}) \tag{6.1}$$

where P_0 is the received power expressed in dBm at a short reference distance d_0.

When transmission power is not known, differences between RSS measures at pairs of receivers can be considered [Liu et al., 2007]. The RSS difference between two nodes provides information about their relative distance from the transmitter and removes the dependency on the actual transmit power.

Although RSS measures are relatively inexpensive and simple to implement in hardware [Patwari et al., 2005], they are susceptible of high errors due to the dynamics of indoor/outdoor environments, mainly due to multipath signals and shadowing. The effect of shadowing is usually modeled as log-normal with standard deviation σ_{dB}. This parameter is relatively constant with distance and typically takes values between 4dB and 12dB depending on the environment considered. This model leads to RSS estimates with variance proportional to their range, i.e., to the distance between the emitter and the node performing the RSS measure.

According to the IEEE 802.22 standard [802.22 WG, 2011], primary users are expected to be several kilometers far away the network and, as a consequence, RSS techniques may not provide enough accuracy in WRAN 802.22 networks.

ToA

In ToA, the distance between the node to be located and an anchor node i is estimated by measuring the signal propagation time between both nodes. In this way, if a signal was sent at time t_1 and reached the receiver node at t_2, the distance d_i between the transmitter and the receiver can be computed as in (6.2), with v_p the propagation velocity. As a consequence, ToA requires precise synchronization between both nodes.

$$d_i = v_p \cdot (t_2 - t_1) \tag{6.2}$$

ToA measures are usually obtained by correlating the received signal with the known transmitted signal [Patwari et al., 2005], a method known as Simple Cross-correlator (SCC). Other estimators such as the Generalized Cross-Correlation (GCC) [Knapp and Carter, 1976] achieve higher performance at the expenses of requiring more knowledge on the signal and noise power.

As in all time-based techniques, measures may be affected by additive noise and multipath signals. Because of this, the accuracy of the method strongly depends on the signal bandwidth, the SNR at reception, and on the ability to estimate the Line of Sight (LoS) signal [Patwari et al., 2005]. Besides, there is another error component due to the clock bias between the emitter and the receiver.

For asynchronous networks where synchronization between the emitter and the receiver cannot be achieved, a common approach is to use two-way ToA measures. In this method, the anchor node transmits a signal to the node to be located, which immediately replies with its own signal. Thus, the time elapsed since the transmission of the first signal until the reception of the replied signal is twice the propagation delay plus a reply delay internal to the node. This internal delay can be measured and subtracted from the total delay to obtain the ToA measure.

Both variants of ToA, however, require the cooperation of the node to be located, either to reply the signal or to achieve synchronization, and therefore none of them can be applied in CRNs to locate a potential attacker.

TDoA

TDoA is based on the difference of the time of arrival of a single signal, transmitted by the node to be located, at two different anchor nodes, thus avoiding the need for synchronization with the emitter. In this case, if a signal was received at time t_1 by the first anchor node and reached the second anchor node at t_2, the difference of distances d_{12} between the transmitter and both receivers is given by (6.3), with v_p the propagation velocity.

$$d_{12} = d_1 - d_2 = v_p \cdot (t_1 - t_2) \tag{6.3}$$

These measures are usually obtained by correlating the signals received at each pair of anchor nodes. Note that contrary to ToA, these measures do not depend on the transmitter clock. Therefore, TDoA can be applied for locating asynchronous transmitters, as it is the case of PUE attackers. As a disadvantage, however, it requires a tight synchronization between each pair of anchor nodes.

A variant of TDoA consists in computing the difference in the time of arrival of two different signals coming from the same emitter, typically a RF signal and an acoustic or ultrasound signal [Savvides et al., 2001]. The distance between both nodes can then be computed as in (6.4), with v_{p1} and v_{p2} the propagation velocity for each type of signal.

$$d = (v_{p1} - v_{p2}) \cdot (t_1 - t_2) \tag{6.4}$$

Different experiments show that this method is extremely accurate, achieving errors of only several centimeters [Priyantha et al., 2000]. However, due to the limited range of acoustic and ultrasound signals, it can only be applied to small and dense networks, e.g., sensor networks. Moreover, it also requires the cooperation of the node to be located in order to reply both signals, as in ToA.

Aside from the error introduced by clock bias, TDoA exhibits the same problems with regard to accuracy as ToA methods, i.e., the main sources of errors are channel noise and multipath fading. However, in some situations where the major source of reflection affects all receivers, the timing error due to multipath may be cancelled [Aatique, 1997], thus achieving higher performance than ToA.

AoA

AoA measures the direction of the received signal at different anchor nodes. There are two common ways of taking AoA measures. The first one requires at least two directional antennas pointed in different directions, such that their main beams overlap. Then, the AoA is estimated based on their RSS ratio, as shown in Figure 6.2.

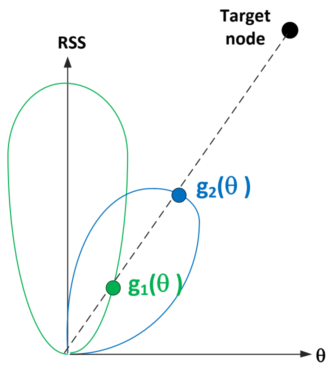


Figure 6.2: AoA measurement

In the second approach, each anchor node makes use of an array of sensors spaced by fractions or a few wavelengths [Pahlavan et al., 2000, Rong and Sichitiu, 2006]. In this case, the AoA is estimated from the differences in the arrival time of the signal at each element of the array.

AoA measures are also affected by additive noise and multipath or shadowing, depending on the approach used. As advantages, either synchronization among anchor nodes or between the emitter and the anchor node are needed. However, it requires multiple antennas elements thus increasing the cost and size of the nodes.

6.1.2 Position Estimation

Based on distance or angle measures provided by a set of anchor nodes, the position of a given emitter is usually estimated by means of methods such as lateration or angulation, depending on whether distances or angles are measured. Whenever the number of observations is reduced to three, then the process is referred to as trilateration for distance measures, or triangulation for angles measures.

A number of other methods exist to estimate the position of a node, such as the bounding box method [Simic and Sastry, 2001] or location fingerprinting [Honkavirta et al., 2009]. The former requires less computations than other schemes at the expenses of providing considerably less accuracy. The latter, consists in storing at a database a set of location signatures, which are built based on the signal characteristics obtained from a set of locations. Then, the position of a given emitter is estimated by comparing its received signal characteristics with those signatures previously catalogued. This proposal poses scalability problems in large networks such as in IEEE 802.22 networks, where potential emitters are rather heterogeneous.

To summarize, the choice of the method has a strong impact on the performance of the location process and depends on the amount of information available and the processor's limitations. In CRNs such as IEEE 802.22 networks, where all CRs are expected to cooperate by reporting sensing data, lateration seems to be the optimal solution. First, spatial diversity provided by CRs can improve the accuracy of the location method. Besides, as the BS can act as a fusion center gathering all data, the computational load of the location process can be exclusively pushed to it.

In the following, we describe how to estimate the position of an emitter by means of angulation and lateration especially focusing on the latter, which is the method selected for our proposal. For the sake of simplicity, we provide the descriptions of these methods in order to obtain the 2-dimensional position of a given emitter. However, all of them can be extended to a 3-dimensional space.

Angulation

In angulation, three or more AoA measures (triangulation or multiangulation) are used to obtain a two-dimension location estimation, which is given by the intersection of the lines of bearing, as shown in Figure 6.3. In theory, direction-based systems require only two anchor nodes to get an estimate: based on the trigonometric proposition that if one side and two angles of a triangle are known, the remaining sides can be computed. Given two anchor nodes with known positions (x_1, y_1) and (x_2, y_2), and their respective AoA measures θ_1 and θ_2, two equations can be easily be derived as in (6.5).

$$\begin{aligned} \tan\theta_1 x - y &= x_1 \tan\theta_1 - y_1 \\ \tan\theta_2 x - y &= x_2 \tan\theta_2 - y_2 \end{aligned} \tag{6.5}$$

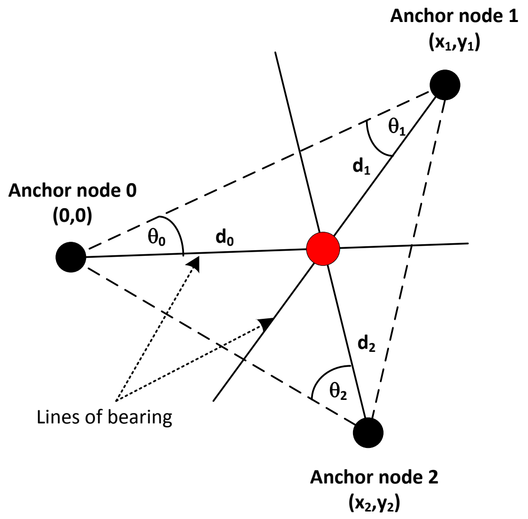


Figure 6.3: Triangulation

By solving this system of equations, the position estimate can be computed as in (6.6).

$$\begin{aligned} x &= \frac{x_2 \tan\theta_2 - x_1 \tan\theta_1 + y_1 - y_2}{\tan\theta_2 - \tan\theta_1} \\ y &= \frac{\tan\theta_1 (x_2 \tan\theta_2 - y_2) - \tan\theta_2 (x_1 \tan\theta_1 - y_1)}{\tan\theta_2 - \tan\theta_1} \end{aligned} \tag{6.6}$$

However, measures are subjected to errors in practice and more than two measures should be employed in order to improve the accuracy of the estimation.

Lateration

This technique is based on measuring distances between the node to be located and three or more anchor nodes with known positions. When distance measures are obtained by means of ToA or RSS, the position is estimated via the intersection of three circles, as depicted in Figure 6.4.

Let (x, y) be the 2-D position of the emitter to be located, which is in the range of 3 anchor nodes whose positions (x_i, y_i) are known. Let d_i be the distance from anchor node i to the emitter, which has been derived from a ToA or RSS measurement. Then, an equation as in (6.7) can be derived for each one.

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{6.7}$$

Lateration based on TDoA, also known as hyperbolic positioning, requires two different measures, i.e., three anchor nodes. For the sake of simplicity, we will assume that one of the anchor

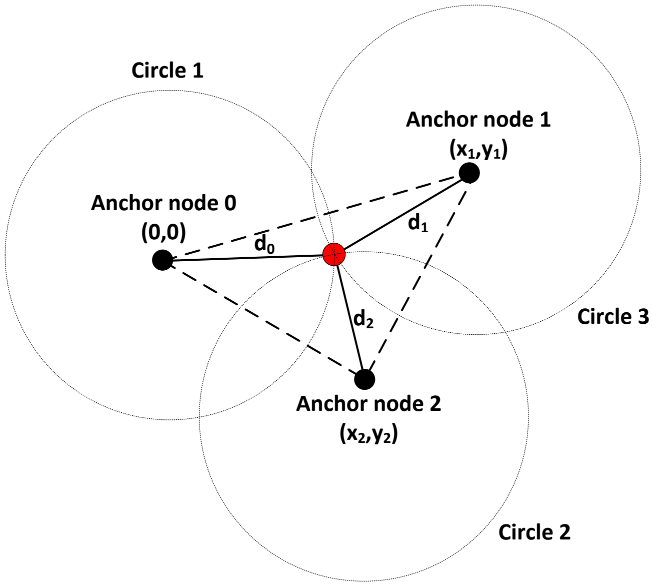


Figure 6.4: Trilateration based on RSS/ToA measures

nodes, which we will denote as reference node, is located at position $(0,0)$. The range distance d_i between the emitter and node i can be expressed as in (6.7). The range difference distance $d_{i,0}$ between node i and the reference node will be given by (6.8).

$$d_{i,0} = d_i - d_0 = \sqrt{(x-x_i)^2 + (y-y_i)^2} - \sqrt{(x)^2 + (y)^2} \tag{6.8}$$

Then, the TDoA measure between the pair receiver i and the reference node, when the propagation velocity is v_p (assuming it remains constant through the paths to all receivers), can be expressed as in (6.9).

$$t_{i,0} = \frac{d_{i.0}}{v_p} = \frac{\sqrt{(x-x_i)^2 + (y-y_i)^2} - \sqrt{(x)^2 + (y)^2}}{v_p} \tag{6.9}$$

Ideally, the intersection of 2 or more different of the above hyperbolas unequivocally provides the emitter position (x, y), as shown in Figure 6.5.

If we apply the same reasoning to a 3-dimensional space, with two receivers at known locations, an emitter can be located onto a hyperboloid. A third receiver provides then a second TDoA measurement and hence locates the emitter on a second hyperboloid. The intersection of both hyperboloids describes a curve on which the emitter lies. If a fourth receiver is now introduced, a third TDoA measurement is available and the intersection of the resulting third hyperboloid with the curve already found with the other three receivers defines a unique point in space.

In order to solve the system of equations provided by the set of RSS/ToA (6.7) or TDoA

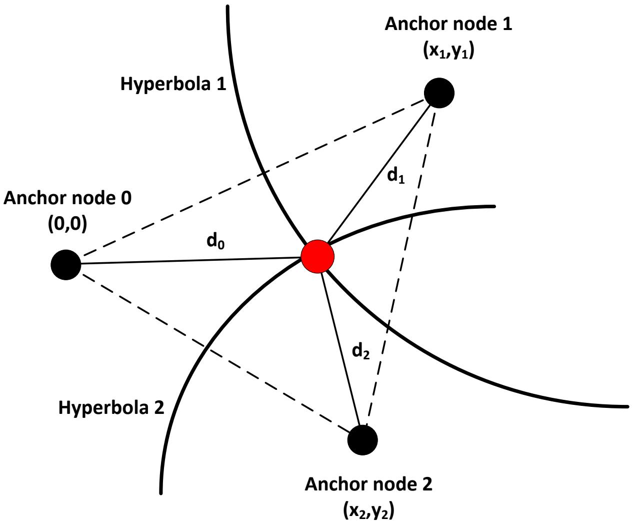


Figure 6.5: Trilateration based on TDoA measures

(6.8) measures, the equations must be first linearized by means of arithmetic operations or other methods such as Taylor's expansion. In the following we describe a method proposed in [Bucher and Misra, 2002]. We detail the mechanism for the two-dimensional location case, and particularized to the equations derived from a set of n TDoA measures.

First, expand equation (6.7) as in (6.10).

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} = \sqrt{x^2 + y^2 + x_i^2 + y_i^2 - 2x_i x - 2y_i y} \tag{6.10}$$

Square both sides of (6.8) as in (6.11).

$$d_i^2 = (d_{i,0} + d_0)^2 \tag{6.11}$$

Expand Equation (6.11), rearrange terms and divide all terms by $d_{i,0}$ as in (6.12).

$$0 = d_{i,0} + 2d_0 + \frac{(d_0^2 - d_i^2)}{d_{i,0}} \tag{6.12}$$

Removing the d_0 term will eliminate all square terms. This can be done by subtracting the equation from receiver 1 from each of the others ($2 \leq i \leq n$) as in (6.13).

$$0 = d_{i,0} - d_{1,0} + \frac{(d_0^2 - d_i^2)}{d_{i,0}} - \frac{(d_0^2 - d_1^2)}{d_{1,0}} \tag{6.13}$$

Recall equation (6.11), expand it and rearrange terms as in (6.14).

$$d_i^2 - d_0^2 = x_i^2 + y_i^2 - 2x_i x - 2y_i y \tag{6.14}$$

Substitute in Equation (6.13) $d_i^2 - d_0^2$ as in (6.15).

$$0 = d_{i,0} - d_{1,0} + \frac{x_i^2 + y_i^2 - 2x_i x - 2y_i y}{d_{i,0}} - \frac{x_1^2 + y_1^2 - 2x_1 x - 2y_1 y}{d_{1,0}} \tag{6.15}$$

A system of linear equations can now be obtained for each receiver $i = 2 \ldots n$ as in (6.16), with a_i, b_i and c_i as in (6.17).

$$0 = a_i x + b_i y + c_i \tag{6.16}$$

$$a_i = \frac{2x_1}{d_{1,0}} - \frac{2x_i}{d_{i,0}} \qquad b_i = \frac{2y_1}{d_{1,0}} - \frac{2y_i}{d_{i,0}}$$

$$c_i = (d_{i,0} - d_{1,0}) + \frac{x_i^2 + y_i^2}{d_{i,0}} - \frac{x_1^2 + y_1^2}{d_{1,0}} \tag{6.17}$$

The set of equations can be written in matrix form as in (6.18), with $\mathbf{A}$,$\mathbf{B}$ and $\mathbf{x}$ as in (6.19).

$$\mathbf{A}\,\mathbf{x} = \mathbf{c} \tag{6.18}$$

$$\mathbf{A} = \begin{pmatrix} a_2 & b_2 \\ a_3 & b_3 \\ \vdots & \vdots \\ a_i & b_i \end{pmatrix} \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} \mathbf{c} = \begin{pmatrix} -c_2 \\ -c_3 \\ \vdots \\ -c_i \end{pmatrix} \tag{6.19}$$

6.1.3 Position estimation with measurement errors

In practice, measures are subjected to errors and the geometrical approaches previously presented lead to position estimation ambiguities. In this case, measures may be expressed as in (6.20), with $\hat{d}_i$ and $\hat{d}_{i,0}$ the distance measures performed by means of RSS/ToA and TDoA respectively, d_i and $d_{i,0}$ the corresponding true distances, and ϵ_i the measure error.

$$\begin{aligned} \hat{d}_i &= d_i + \varepsilon_i \\ \hat{d}_{i,0} &= d_{i,0} + \varepsilon_i \end{aligned} \tag{6.20}$$

The effect of noisy measures, from the geometric point of view, is that the set of circles defined by equations in (6.7) or the hyperbolas defined as in (6.9) rarely intersect.

The set of measures leads now to a system of equations with no unique solution which can be expressed as in (6.21), with $\mathbf{A}$ and $\mathbf{B}$ as in (6.19), $\hat{\mathbf{x}}$ the position estimate as in (6.22) and $\mathbf{e}$ a vector containing the error performed in each measure.

$$\mathbf{A}\,\hat{\mathbf{x}} = \mathbf{c} + \mathbf{e} \tag{6.21}$$

$$\hat{\mathbf{x}} = \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \tag{6.22}$$

Then, the position estimate can be approximated by means of heuristic approaches which try to minimize the error performed in the estimation. Generally speaking, these approaches can be classified into non-iterative methods, iterative methods and filtering methods. Usually, iterative methods provide more accurate estimations but require initial values of the unknown parameters. These initial values can be obtained, for example, by using a non-iterative method. Then, the estimate is updated at each iteration until some predefined threshold is crossed. On the other hand, filtering methods are usually applied when the node to be located is moving at a certain speed along a specific route.

In the following, we describe the most representative methods for each category: the Least Squares (LS) method as an example of non-iterative methods, Taylor series estimation as an iterative method and the extended Linear Kalman Filter (LKF) as a filtering method.

Least Squares method

This method gives the solution which minimizes the sum of deviations squared, i.e., the least square error, for the set of measures. This solution is optimal provided that measure errors are uncorrelated and Gaussian distributed, with zero mean and equal variances. Consider the matrix form of the set of equations in (6.21) and let S be the sum of square errors, which we aim to minimize, as in (6.23).

$$\mathbf{S} = \sum_{i=1}^{n} e_i^2 = \mathbf{e}^{\mathrm{T}}\mathbf{e} = (\mathbf{A}\hat{\mathbf{x}} - \mathbf{c})^{\mathrm{T}}(\mathbf{A}\hat{\mathbf{x}} - \mathbf{c}) \tag{6.23}$$

By setting the gradient of S to zero as in (6.24), the solution is found as in (6.25).

$$\frac{\partial \mathbf{S}}{\partial \hat{\mathbf{x}}} = \frac{\partial(\hat{\mathbf{x}}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\hat{\mathbf{x}} - \hat{\mathbf{x}}^{\mathrm{T}}\mathbf{A}\mathbf{c} + \mathbf{c}\mathbf{c}^{\mathrm{T}} - \mathbf{c}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}} = -2\mathbf{A}^{\mathrm{T}}\mathbf{c} + 2\mathbf{A}^{\mathrm{T}}\mathbf{A}\hat{\mathbf{x}} = 0 \tag{6.24}$$

$$\hat{\mathbf{x}} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{c} \tag{6.25}$$

When not all measures exhibit the same reliability, a Weighted Least Squares (WLS) method can be applied. In such case, the target is to minimize the expression as in (6.26), where w_i is the weight assigned to each observation that reflects the uncertainty of the measurement, and W the matrix of weights.

$$\mathbf{S} = \sum_{i=1}^{n} w_i e_i^2 = \mathbf{e}^{\mathrm{T}}\mathbf{W}\mathbf{e} = (\mathbf{A}\hat{\mathbf{x}} - \mathbf{c})^{\mathrm{T}}\mathbf{W}(\mathbf{A}\hat{\mathbf{x}} - \mathbf{c}) \tag{6.26}$$

The solution that minimizes the sum of squares errors is then given by (6.27).

$$\hat{\mathbf{x}} = (\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{c} \tag{6.27}$$

Taylor's series estimation

Taylor-series method [Foy, 1976] is an iterative scheme which requires an initial position estimate. The system of equations derived from the range measures are linearized by expanding them in Taylor series at a point, i.e., at the initial estimate, and keeping only terms below the second order. At each iteration, the algorithm improves the estimate by determining the local linear LS solution until a given criterion is satisfied.

Let $f_i(x, y)$ be the range measurement performed by node i, and (x_v, y_v) a guess of the true position of the emitter. Then,

$$x = x_v + \delta_x \qquad y = y_v + \delta_y$$

where (δ_x, δ_y) is the estimation error.

Expand $f_i(x, y)$ in Taylor series around (x_v, y_v) as in (6.28), keeping only the first order coefficients a_{ix} and a_{iy} obtained as in (6.29).

$$f_i(x_v, y_v) + a_{ix}\delta_x + a_{iy}\delta_y \tag{6.28}$$

$$a_{ix} = \left.\frac{\partial f_i(x, y)}{\partial x}\right|_{x=x_v} \qquad a_{iy} = \left.\frac{\partial f_i(x, y)}{\partial y}\right|_{y=y_v} \tag{6.29}$$

Then, for each anchor node $1 \leq i \leq n$ we have a linear equations as in (6.30)

$$\hat{f}_i(x_v, y_v) + a_{ix}\delta_x + a_{iy}\delta_y = \hat{d}_{i,0} + \varepsilon_i \tag{6.30}$$

where $\hat{d}_{i,0}$ is the range difference distance measured, and ε_i includes the TDoA measurement error and the linearization error.

According to this, the matrix representation of the linear system of equations is as in (6.31) with vectors and matrices defined as in (6.32).

$$\mathbf{A} \times \boldsymbol{\delta} = \mathbf{z} + \varepsilon \tag{6.31}$$

$$\mathbf{A} = \begin{pmatrix} a_{1x} & a_{1y} \\ a_{2x} & a_{2y} \\ \vdots & \vdots \\ a_{nx} & a_{ny} \end{pmatrix} \boldsymbol{\delta} = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix}$$

$$\mathbf{z} = \begin{pmatrix} \hat{d}_{1,0} - f_1(x_v, y_v) \\ \hat{d}_{2,0} - f_2(x_v, y_v) \\ \vdots \\ \hat{d}_{n,0} - f_n(x_v, y_v) \end{pmatrix} \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix} \tag{6.32}$$

Assuming that $\mathbf{A}$ is full rank, the value of $\boldsymbol{\delta}$ that gives the least-sum-squared error can be computed as in (6.33).

$$\boldsymbol{\delta} = [\mathbf{A}^{\mathrm{T}}\mathbf{A}]^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{z} \tag{6.33}$$

The target is to iterate the method until the components of $\boldsymbol{\delta}$ are below a given threshold. Therefore, in every iteration a new initial guess of the emitter's location (x'_v, y'_v) is derived from the previous values of δ_x and δ_y as in (6.34). Under convergence conditions, $\boldsymbol{\delta}$ gets closer to a zero vector with each iteration.

$$x'_v = x_v + \delta_x \quad y'_v = y_v + \delta_y \tag{6.34}$$

This method is robust and provide accurate estimation, and can make use of redundant measures. Its main drawback, however, is that its convergence strongly depends on the selection of the initial guess.

Extended Kalman's filter

When the node to be located is moving, the motion characteristics can be employed in the location process. Filtering techniques such as the Extended Kalman Filter (EKF) [Brown and Hwang, 1997] have been widely used for tracking moving objects. The EKF is an extension of the LKF [Kalman et al., 1960] where the system model is linearized. It provides reasonable performance but, unlike its linear version, it is not in general an optimal estimator: if the initial estimate of the state is wrong or if the process is modeled incorrectly, the filter may quickly diverge.

Let us consider a distance measurement as in (6.35), where i indexes the node performing the measurement and k the time instants, $d_i^{(k)}$ is the true distance given by (6.36) and $\varepsilon_i^{(k)}$ is the measure error.

$$\hat{d}_i^{(k)} = d_i^{(k)} + \varepsilon_i^{(k)} \tag{6.35}$$

$$\hat{d}_i^{(k)} = \sqrt{(x - x_i)^2 + (y - y_i)^2} - \sqrt{(x)^2 + (y)^2} \tag{6.36}$$

Given the position of the mobile node (x^k, y^k), which varies over time, and the velocity components along each coordinate (v_x^k, v_y^k), let us define the state vector at time instant k as in (6.37).

$$\boldsymbol{\theta}^{(k)} = [x^{(k)}, y^{(k)}, v_x^{(k)}, v_y^{(k)}]^{\mathrm{T}} \tag{6.37}$$

Then, the system dynamic model is given by (6.38), where $\mathbf{\Phi}^{(k-1)}$ and $\mathbf{B}^{(k-1)}$ are as in (6.39), δ_t is the sampling time increment, $\boldsymbol{\varepsilon}^{(k-1)}$ is the acceleration noise modeled as a white Gaussian random vector of zero mean and covariance matrix $\mathbf{Q}^{(k-1)}$.

$$\boldsymbol{\theta}^{(k)} = \mathbf{\Phi}^{(k-1)}\boldsymbol{\theta}^{(k-1)} + \mathbf{B}^{(k-1)}\boldsymbol{\varepsilon}^{(k-1)} \tag{6.38}$$

$$\mathbf{\Phi}^{(k-1)} = \begin{pmatrix} 1 & 0 & \delta_t & 0 \\ 0 & 1 & 0 & \delta_t \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{B}^{(k-1)} = \begin{pmatrix} 0.5\delta_t^2 & 0 \\ 0 & 0.5\delta_t^2 \\ \delta_t & 0 \\ 0 & \delta_t \end{pmatrix} \tag{6.39}$$

Let us denote the observation vector at instant k as in (6.40).

$$\mathbf{p}^{(\mathbf{k})} = [x^k, y^k]^{\mathrm{T}} \tag{6.40}$$

Then, the measurement model becomes as in (6.41), with $\mathbf{n}^{(k)}$ the noise vector, modeled as white Gaussian random vector of zero mean and covariance matrix denoted by $\mathbf{Q}^{(k-1)}$, and with $\mathbf{g}^{(k)}\boldsymbol{\theta}^{(k)}$ as in (6.42).

$$\mathbf{p}^{(\mathbf{k})} = \mathbf{g}^{(k)}\boldsymbol{\theta}^{(k)} + \mathbf{n}^{(k)} \tag{6.41}$$

$$\mathbf{g}^{(k)}\boldsymbol{\theta}^{(k)} = [d_1^{(k)} d_2^{(k)} \cdots d_n^{(k)}] \tag{6.42}$$

Equation (6.41) is non-linear and therefore the LKF cannot be applied directly. However, it can be approximated as in (6.43), where $\hat{\theta}^{(k|k-1)}$ is the estimate of the state vector at time instant k based on the previous estimates and observations including that at time $k-1$ and $\mathbf{G}^{(k)}$ as in (6.44).

$$\mathbf{p}^{(\mathbf{k})} = \mathbf{G}^{(k)}\boldsymbol{\theta}^{(k)} + \mathbf{n}^{(k)} + [\mathbf{g}^{(k)}\hat{\boldsymbol{\theta}}^{(k|k-1)} - \mathbf{G}^{(k)}\hat{\boldsymbol{\theta}}^{(k|k-1)}] \tag{6.43}$$

$$\mathbf{G}^{(k)} = \left.\frac{\partial \mathbf{g}^{(k)}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{k|k-1}} \tag{6.44}$$

At each step, the prediction, Kalman gain computation and estimate updating are performed sequentially. The state estimate is predicted as in (6.45).

$$\hat{\boldsymbol{\theta}}^{k|k-1} = \mathbf{\Phi}\hat{\boldsymbol{\theta}}^{k-1|k-1} \tag{6.45}$$

The error covariance matrix is predicted as in (6.46) and the Kalman gain matrix as in (6.47).

$$\mathbf{C}^{(k|k-1)} = \mathbf{\Phi}\mathbf{C}^{(k-1|k-1)}\mathbf{\Phi}^{\mathrm{T}} + \mathbf{B}\mathbf{Q}\mathbf{B}^{\mathrm{T}} \tag{6.46}$$

$$\mathbf{K}^{(k)} = \mathbf{C}^{(k|k-1)}(\mathbf{G}^{(k)})^{\mathrm{T}}(\mathbf{G}^{(k)}\mathbf{C}^{(k|k-1)}\mathbf{G}^{(\mathbf{k})})^{\mathrm{T}} + \mathbf{R})^{-1} \tag{6.47}$$

Finally, the state estimate and the covariance matrix are updated according to (6.48) and (6.49), respectively.

$$\hat{\boldsymbol{\theta}}^{(k|k)} = \hat{\boldsymbol{\theta}}^{(k|k-1)} + \mathbf{K}^{(\mathbf{k})}(\mathbf{p}^{(k)} - \mathbf{G}^{(k)}\hat{\boldsymbol{\theta}}^{(k|k-1)}) \tag{6.48}$$

$$\mathbf{C}^{(k|k)} = (\mathbf{I} - \mathbf{K}^{(\mathbf{k})}\mathbf{G}^{(k)})\mathbf{C}^{(k|k-1)} \tag{6.49}$$

Note that this method requires an initial estimate $\hat{\theta}^{(0|0)x}$ and of its covariance matrix $\mathbf{C}^{(0|0)}$. The initial velocities can be set to zero if there is no additional information, and the initial position estimate can be obtained by using a non-iterative method. On the other hand, the measurement error covariance matrix R can be estimated based on the known accuracy of the distance measures.

6.2 Proposal of a cooperative location method for detection of PUE attacks

In this section, we present a cooperative location method to estimate the position of an emitter and detect PUE attacks in CRNs. The proposed method is intended to be applied to infrastructure-based CRNs based on the 802.22 WRAN standard, where primary users are TV towers.

In order to detect primary emitters in the CRN channel of operation, either legitimate or fake, all CRs perform spectrum sensing and report their measures to the BS. The BS acts as data fusion center and takes a decision about the existence of a primary transmission based on these reports. If there is evidence of the presence of a primary user, the CRN applies a localization method to estimate the position of the emitter. The transmission is considered to be legitimate whenever its position matches any of the known TV transmitters' positions. Otherwise, it is assumed that a PUE attack is being performed.

The location method is based on the TDoA technique described in Section 6.1.1 and applies multilateration in order to estimate the position of the emitter. In particular, Series-Taylor estimation is used to solve the system of equations derived from the set of TDoA measures. On the one hand, TDoA does not require the collaboration of the node to be located and provides higher accuracy than RSS. On the other hand, Series-Taylor estimation is a commonly used approach for node's location because it provides higher accuracy than most of the non-iterative methods. Besides, it is preferable to filtering schemes such as the Extended Kalman's filter which exhibit higher complexity, and provide no significantly improvement when the node to be located is static.

In the following, we provide a detailed description of this method. We first present the assumptions that have been considered for its design. Next, we describe the steps to follow in order to obtain TDoA measures and to apply Series-Taylor estimation in order to derive the position of the emitter.

6.2.1 Assumptions

In order to develop the proposed method, the following assumptions have been adopted:

- CRs are randomly located and their position is fixed and known to the BS.
- The BS has a database with the true position of TV primary emitters and their respective frequencies of operation.
- The PUE attacker remains at a fixed position and can have different capabilities regarding power transmission.
- The BS is responsible for scheduling quiet periods with sensing purposes, in which no station is allowed to transmit in the channel to be sensed. As recommended by the IEEE 802.22 WRAN standard, two different types of sensing mechanisms are used: fast sensing, which can be performed frequently and generally relies on energy-based sensing techniques, and fine sensing, which is based on more complex techniques that allow obtaining specific features of the signal but require a longer observation time. As we will further explain, the proposed localization method will be embedded in the fine sensing process.
- The measures performed by each node are normally distributed with zero mean [Gezici et al., 2005] and statistically independent. The latter can be assumed if CRs are distanced more than a few hundred of meters [Min et al., 2010].

6.2.2 Obtaining TDoA measures

When the BS detects the presence of a potential primary emitter, it requests a set of n nodes to record the signal transmitted by the primary emitter, and send this record back to it. With these data, the BS derives the set of TDoA measures by correlating the signal received by itself with the records received from the CRs. The accuracy of these measures has a strong impact on the position estimation and highly depends on tight synchronization between the CRs and the BS. Although many synchronization proposals on the state-of-the-art [Pandey and Agrawal, 2006] can be applied to CRNs, their accuracy get up to 1μs, which could lead to errors of hundreds of meters. For this reason, in the following we propose a method to synchronize the recordings from several nodes. The chosen solution is fairly straight forward and can be summarized in five steps.

1. When the localization process starts, the BS requests all the nodes in the CRN to record the potential primary signal. In its turn, the BS makes its own recording.
2. While every node in the network is recording the signal, the BS sends a marker signal, i.e., an especial signal known to every CR. Each CR will add the received marker signal to its recording and will send it to the BS.
3. The BS synchronizes its own recording with those provided by the set of CRs. Since the BS knows the position of every CRN node, it can compute the time elapsed since the transmission of the marker signal until it is received at each node. Assuming that the BS is placed in $(0, 0)$, this time can be computed as in (6.50) with i the CR receiving the marker signal, (x_i, y_i) its position and v_p the propagation velocity.

$$\Delta t_i = \frac{\sqrt{x_i^2 + y_i^2}}{v_p} \tag{6.50}$$

4. When the recordings have been synchronized, the BS derives each TDoA measure by correlating its own recording with the one provided by every CR.

Figure 6.6 depicts this process. The signal on the top represents the recording made by the BS. Below, the recordings for two different anchor nodes are shown. The fragments of signal framed in red represent the same portion of signal but time-shifted, since every node receives the signal at different instants of time. Assuming that the BS sends the marker signal at instant $t = 0$, the anchor nodes will receive this signal at instant $t = \Delta t_i$. Then, the TDoA measure for each anchor node can be computed as in

$$\tau_i = t_i - t_0 \tag{6.51}$$

with t_i and t_0, the instant at which anchor node i and the BS receive the red framed fragment of signal, according to the BS clock. Note that, the value of t_i can be computed by adding up to Δt_i the time difference between the instant of reception of the marker signal at the anchor node and the beginning of the red framed fragment.

6.2.3 Applying Taylor-series estimation for position estimation

As seen in Section 6.1.1, a TDoA measure τ_i is obtained for each pair anchor node i with $i = 1..n$. From these measures, the BS estimates the position of the emitter by means of Taylor-series estimation as follows. Assuming that the BS is placed at the origin $(0, 0)$, a difference range distance can be estimated as in (6.52), with Δ_i the TDoA measurement error, $\varepsilon_i = \Delta_i v_p$ the distance error, and with $d_{i,0}$ the true value of the difference distance, which can be expressed as as a function of the emitter coordinates as in (6.53).

$$d_{i,0} = \hat{d}_{i,0} + \varepsilon_i = \tau_i v_p = t_{i,0} v_p + \Delta_i v_p \tag{6.52}$$

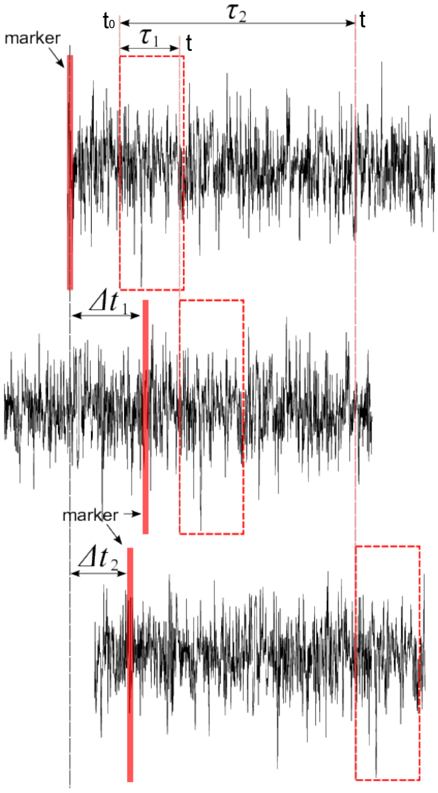


Figure 6.6: Obtaining TDoA measures from synchronized recordings

$$d_{i,0} = f_i(x,y) = \sqrt{(x - x_i)^2 + (y - y_i)^2} - \sqrt{(x)^2 + (y)^2} \tag{6.53}$$

The system of non-linear equations to be solved is then given by (6.54).

$$\hat{d}_{i,0} = \sqrt{(x - x_i)^2 + (y - y_i)^2} - \sqrt{(x)^2 + (y)^2} + \varepsilon_i \tag{6.54}$$

As mentioned in Section 6.1.3, Taylor series estimation is an iterative scheme which requires an initial guess of the location of the emitter denoted by (x_v, y_v). The set of equations given by (6.54) are linearized by applying Taylor series expansion around the initial guess as in (6.55), where a_{ix} and a_{iy} are the first order terms computed as in (6.57).

$$f_i(x_v, y_v) + a_{ix}\delta_x + a_{iy}\delta_y = \hat{d}_{i,0} + \varepsilon_i \tag{6.55}$$

$$a_{ix} = \frac{x_v - x_0}{\sqrt{(x_v - x_0)^2 + (y_v - y_0)^2}} - \frac{x_v - x_i}{\sqrt{(x_v - x_i)^2 + (y_v - y_i)^2}} \tag{6.56}$$

$$a_{yx} = \frac{y_v - y_0}{\sqrt{(x_v - x_0)^2 + (y_v - y_0)^2}} - \frac{y_v - y_i}{\sqrt{(x_v - x_i)^2 + (y_v - y_i)^2}} \tag{6.57}$$

The solution to this linear system of equations can be obtained by means of a LS method as in (6.33). With this approach, the sum of squared errors performed in the estimation is minimized. Nevertheless, since every anchor node is subjected to different environment conditions and thus have different SNRs, not all the measures exhibit the same reliability. Thus, in order to assign a higher weight to those anchor nodes experimenting best conditions with regard to the reception of the signal, we applied a WLS method. As a result, the target is to minimize the expression given in (6.58).

$$\sum_{i=1}^{n} w_{ii}\varepsilon_i^2 = \boldsymbol{\varepsilon}^{\mathrm{T}}\mathbf{W}\boldsymbol{\varepsilon} = (\mathbf{A}\boldsymbol{\delta} + \boldsymbol{\varepsilon})^{\mathrm{T}}\mathbf{W}(\mathbf{A}\boldsymbol{\delta} + \boldsymbol{\varepsilon}) \tag{6.58}$$

Analogously to (6.27), the solution that minimizes this expression is given by (6.59), where $\mathbf{W}$ is the error covariance matrix.

$$\boldsymbol{\delta} = [\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{A}]^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{z} \tag{6.59}$$

At each new iteration, a new guess of the emitter's location is derived as in (6.60).

$$x_v = x_v + \delta_x \quad\quad y_v = y_v + \delta_y \tag{6.60}$$

In order to provide high accuracy in the position estimation, a new iteration is performed until the components of δ are below a given threshold or when there is evidence of lack of convergence, i.e., when after a given number of iterations the error does no tend to zero. According to the position estimated in the last iteration, the BS will take a decision about the existence of a primary user or an attacker

This method requires to estimate the error covariance matrix $\mathbf{W}$ and to make an initial guess of the emitter position (x_v, y_v) for the first iteration. In the following, we describe how to derive both parameters.

Computing the covariance matrix W

Assuming that the measures performed by the set of nodes are statistically independent, the error covariance matrix W is a diagonal matrix where its non-zero elements w_{ii} represent the weights assigned to each TDoA measure. These weights can be computed as in (6.61), with σ_i^2 the error variance of the TDoA measurement. In this way, measures exhibiting lower errors have also a lower impact on the position estimation.

$$w_{ii} = \frac{1}{\sigma_i^2} \tag{6.61}$$

TDoA measures are computed as the difference of two ToA measures between a node i and the BS. In the literature, the error performed in ToA measures is usually modeled as a random variable following a normal distribution, and its variance is derived by means of the Cramer-Rao Lower Bound (CRLB) [Gezici et al., 2005], which provides a lower bound on the variance in a multipath-free channel. Therefore, TDoA errors can be modeled as normal random variables with variance $\sigma_i^2 + \sigma_0^2$, with σ_0 the variance of the BS measure and σ_i the variance of the measure reported by anchor node i as in (6.62)

$$\sigma_i^2 \geq \frac{1}{8\pi^2 \cdot B^2 \cdot SNR_i} \tag{6.62}$$

where B is the signal bandwidth and SNR$_i$ the signal-to-noise relation at anchor node i.

In order to derive the SNR for every anchor node, the Okumura-Hata model for suburban areas is used for path loss calculation [Eksim et al., 2009], as recommended by the IEEE 802.22 standard. As a result, the SNR$_i$ at a given anchor node i can be expressed as in (6.63), with ΔL_p the path losses as in (6.64), SNR$_0$ the signal-to-noise ratio at the BS, d_i and d_0 the distances from the emitter to anchor node i and the BS respectively, and h_b the current altitude of the emitter to be located.

$$SNR_i(dB) = SNR_0(dB) - \Delta L_p(dB) \tag{6.63}$$

$$\Delta L_p(dB) = [44.9 - 6.55 \cdot \log_{10}(h_b)] \log \frac{d_i}{d_0} \tag{6.64}$$

Selection of the initial guess

The initial guess (x_v, y_v) plays an important role in the performance of the algorithm. In fact, one of the disadvantages of the algorithm is that it may not converge depending on the goodness of such guess. In such case, however, another attempt with a different initial guess can be made. As the initial estimation of the emitter's position (x_v, y_v) is updated at the end of each iteration as in (6.34), lack of convergence can be detected either by verifying that the estimated position is further than a given distance, i.e., the maximum receiving distance, or that the error $(\delta_{x_{max}}, \delta_{y_{max}})$ does not get closer to zero after a new iteration.

Aiming to minimize the number of iterations needed and to avoid the convergence problem, we propose a method for properly selecting the initial guess. In this method, the BS precomputes and stores at a database a m-by-n matrix of TDoA measures, with m the number of potential positions of emitters and n the number of CRs. p out of the m positions match the position of real primaries. The remaining positions are obtained by drawing a grid on the area and taking the central point of every single cell, as shown in Figure 6.7.

Then, when a potential primary user must be identified, we apply the following steps:

- Compute the Euclidean distance between a vector containing the TDoA measures performed by the n stations and each of the m vectors (rows of the matrix) containing the TDoA the precomputed measures. Each vector m corresponds to a potential position of the emitter.

- Create a vector V containing the m positions sorted according to the distance computed in the previous point, in ascending order.

- Select as initial guess the first element of V and apply the WLS algorithm.

  - If the algorithm does not converge, select the following element of V as initial guess and apply again the WLS algorithm.

  - If the algorithm converges, it provides the final estimation of the position of the emitter.

Intuitively, the first element of V, which represents the position that minimizes the distance between the reported TDoA measures and the precomputed TDoA measures, should be the closest one to the real position of the emitter. In practice, it may not always be so, due to errors in the measures reported by CRs. However, this does not represent a serious problem since, if the algorithm does not convergence because a wrong initial guess has been selected, the location process can be performed again by selecting the next element of V. Note that the smaller the size of the cells defined in the grid, the more accuracy in the estimation. However, this comes at the expenses of maintaining a larger database, and therefore a higher cost in terms of allocated memory and computation time. With the purpose of avoiding an unnecessary waste of resources, smaller cells could be defined in those areas where a higher precision is required, and keep larger cells in the remaining areas.

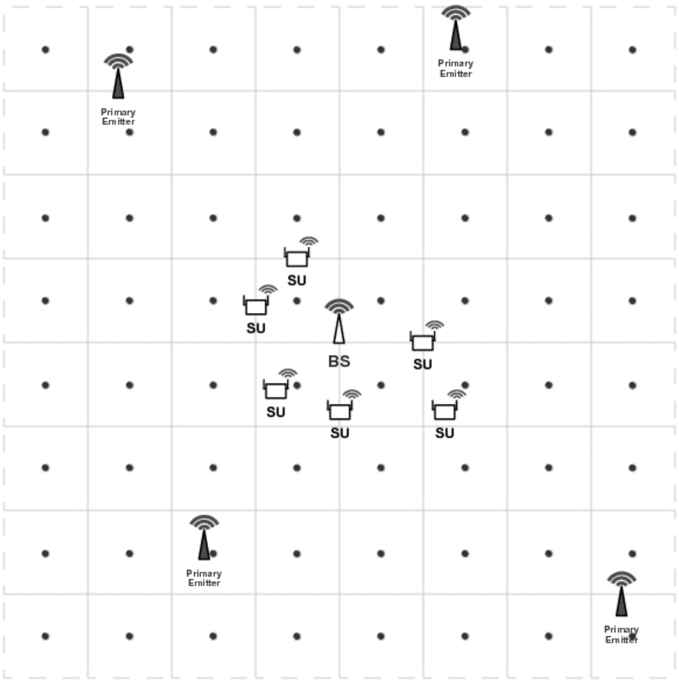


Figure 6.7: Positions for TDoA measures in database

6.3 Performance evaluation

In this section we evaluate the goodness of the proposed method which has been programmed in Matlab [MATLAB, 2009]. Simulation results for different scenarios are provided.

6.3.1 Methodology

As explained in Section 6.2.2, the members of the CRN must record a fragment of the potential primary signal and send it to the BS. During the sensing period in which recordings are being performed, the channel cannot be used for data transmission so as to avoid interferences. When this period end, CRs use the channel to send their recordings, meaning that for a given period of time the channel is not available for normal operation. Because this can introduce a significant overhead in terms of control data transmission and lost of capacity of the CRN, in this work we have analyzed the amount of extra time needed to carry on the location process. The results of this analysis are shown in Section 6.3.4.

On the other hand, the efficiency of the proposed method in detecting PUE attacks relies on the accuracy of the estimated position of the emitter. Thus, in order to prove the goodness of our proposal, we have also evaluated it in terms of location error. The analysis of the accuracy of a location method can be performed by mean of several parameters [Yu et al., 2009]. Among them, the Root-Mean-Square Error (RMSE) is one of the most widespread and is computed as follows. Let us consider n estimations of the position of a given emitter denoted by $(\hat{x}_i, \hat{y}_i)$ with $1 \leq i \leq n$, and (x, y) the true position of the emitter. Then, the RMSE of the location estimation is given by (6.65).

$$RMSE = \sum_{i=1}^{n} \sqrt{\frac{1}{n}(\hat{x}_i - x)^2 + (\hat{y}_i - y)^2} \tag{6.65}$$

However, as pointed out in [Yu et al., 2009], caution is needed when estimating this parameter. Consider the case in which a set of accurate samples are obtained, except for some abnormal samples which are extremely large compared to the rest. If these samples are taken into account to compute the RMSE, the obtained value will probably be larger than it should. As a consequence, it is often necessary to exclude these abnormal samples from the set.

Another parameter to measure accuracy is the Cumulative Distribution Probability (CDP). The CDP allows to easily obtain the percentage of samples exhibiting an error below a given value, where the error is typically defined as the Euclidean distance between the true location and the estimated location. From our point of view, this is the optimal parameter to evaluate our method since the probability of PUE detection can be directly derive from it. As mentioned in Section 1.1.4, in IEEE 802.22 networks the probability of detection and miss detection of true primary users must be above 0.9 and below 0.1, respectively. Because of this, our goal is to achieve the necessary location accuracy in order to fulfill this requirement. In the following,

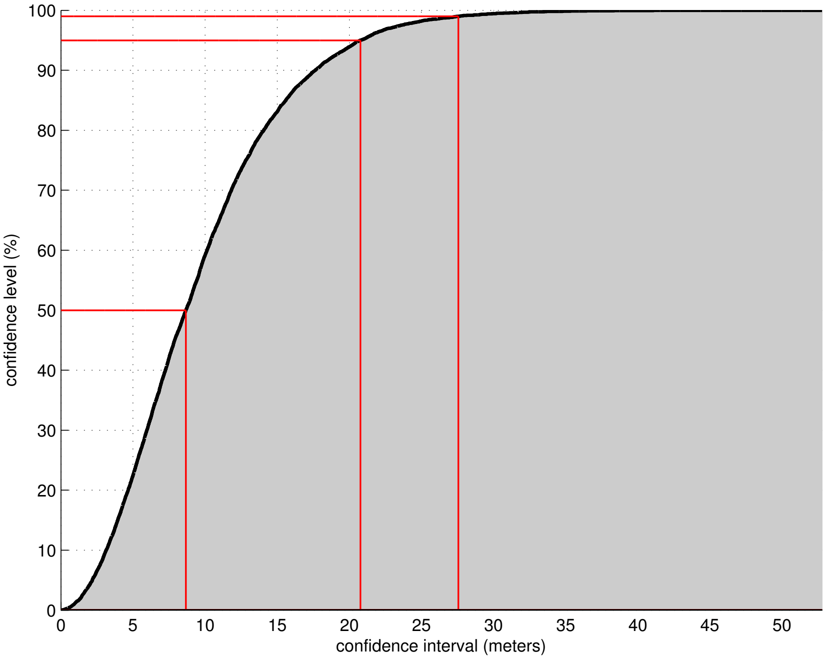


Figure 6.8: Normalized cumulative histogram (10000 experiments),(x,y)=(8000,1000), SNR=-10dB, n=100

we provide a description of the methodology followed in order to derive the CDP of the location error performed with the proposed method.

The error performed in the estimation across different experiments is computed as in (6.66), with (x_i, y_i) the estimation of the position at the i-th experiment and (x, y) the real position of the emitter.

$$e_x^i = \hat{x}_i - x \qquad e_y^i = \hat{y}_i - y \qquad e^i = \sqrt{{e_x^i}^2 + {e_y^i}^2} \tag{6.66}$$

From the set of error samples, we derive the normalized cumulative histogram and compute different confidence intervals. Figure 6.8 represents the normalized cumulative histogram for 10000 experiments when the emitter is placed at $(8000, 1000)$, with the coordinates expressed in meters. The average SNR at reception for the BS is of -10dB and the number of CRs is $n = 100$. The 50%, 95% and 99% confidence intervals are represented with red lines and correspond to distance errors of approximately 8.66, 20.76 and 27.56 meters respectively. Notice that under these conditions, with a probability of 0.99 the location error is at most a $\frac{27.56}{\sqrt{(8000)^2+1000^2}} \approx 0.34\%$ of the distance between the emitter and the BS. Figure 6.9 shows the mean distance error and confidence intervals for a varying number of cooperating stations ranging from 10 to 100 with the same conditions as in Figure 6.8. The figure clearly shows that cooperation actually improves the estimation of the emitter's position, leading in the worst case ($n = 10$) to a distance error of roughly 450.7 meters and in the best case ($n = 100$) to a distance error approximately of 9.07 meters with a level of confidence of 99%.

By plotting the error samples on a 2-dimensional Cartesian coordinate system as in Figure 6.10,

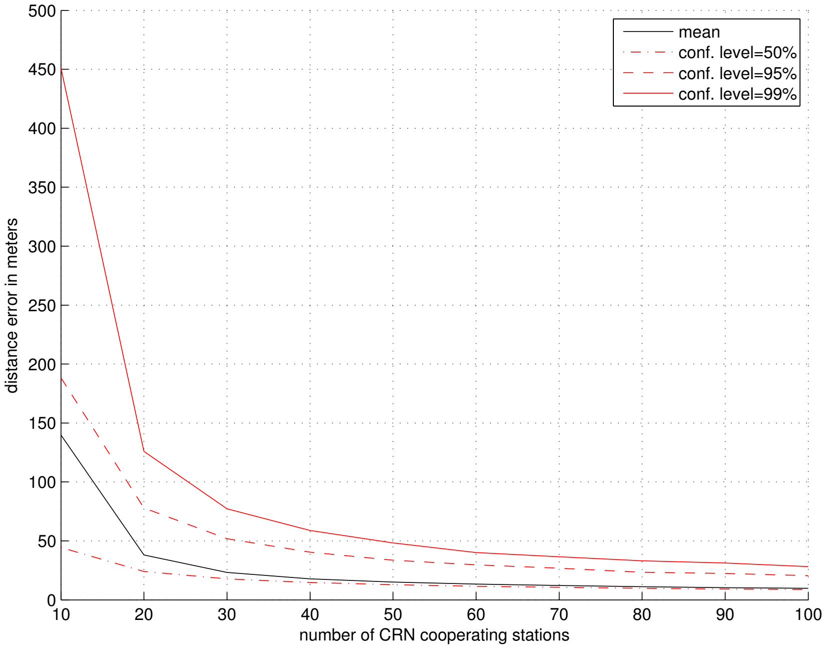


Figure 6.9: 99% confidence interval error (10000 experiments), (x,y)=(8000,1000), SNR=-10dB, n=100

it can be observed that error samples spread over an elliptical area where the center of the ellipse is the real position of the emitter. Therefore, confidence levels can be also represented as an ellipse containing the α% of the samples, with α the level of confidence. Figure 6.10 represents the location error for an attacker placed at $(8000, 1000)$ for 1000 experiments with a SNR of -10dB, and $n = 100$ cooperating stations. The orientation of the ellipse is given by the position of the attacker (x, y) and can be computed as in (6.67).

$$\theta = \arctan \frac{y}{x} \tag{6.67}$$

In order to obtain the eccentricity of the ellipse, we project every error sample ($e_x^i = x^i - x, e_y^i = y^i - y$) onto the major and minor ellipse axis as in (6.68). Let E'_x be the vector with the error values e'^i_x projected onto the major axis and E'_y the vector with the projections onto the minor axis e'^i_y. Then, we can compute the eccentricity as in (6.69), with MAD(E'_x) and MAD(E'_y) the mean absolute deviation of the error in each of the projected axis obtained as in (6.70).

$$e'^i_x = e^i \cos(\theta) \qquad e'^i_y = e^i \sin(\theta) \tag{6.68}$$

$$\varepsilon = \sqrt{1 - \left(\frac{\mathrm{MAD}(E'_y)}{\mathrm{MAD}(E'_x)}\right)^2} \tag{6.69}$$

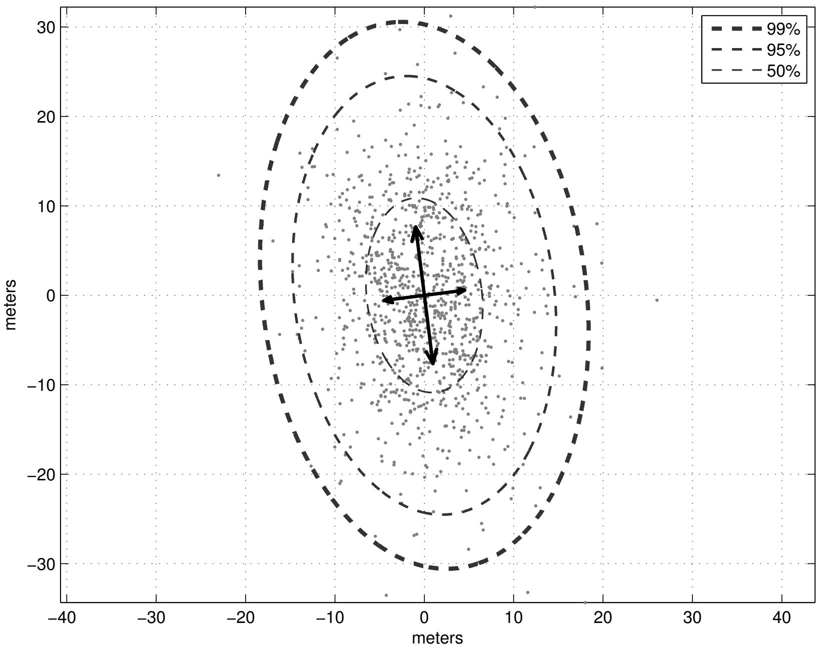


Figure 6.10: 99% confidence interval error (1000 experiments), (x,y)=(8000,1000), SNR=-10dB, n=100

$$\mathrm{MAD}(E'_x) = \frac{1}{n}\sum_{i=1}^{n} |e'^{i}_{x}| \quad \mathrm{MAD}(E'_y) = \frac{1}{n}\sum_{i=1}^{n} |e'^{i}_{y}| \tag{6.70}$$

Once the eccentricity of the ellipse has been computed, the ellipse containing a given percentage of the samples can be easily obtained. Ellipses in Figure 6.10 represent the 50%, 95% and 99% confidence intervals. $\mathrm{MAD}(E'_x) \approx 4.54$ meters and $\mathrm{MAD}(E'_y) \approx 7.7$ meters are represented as arrows centered at the emitter position. As it can be seen, in the 99% of the samples the estimation error is limited to only 30.55 meters on the major axis (worst case) which represents an error of just a $\frac{30.55}{\sqrt{8000^2+1000^2}} \approx 0.37\%$ of the distance between the emitter and the BS.

6.3.2 Simulation scenario

In order to evaluate the performance of the proposed location method, we have considered a CRN following the IEEE 802.22 WRAN standard, which is composed by a BS located at the origin and a set of CR nodes uniformly distributed within a square area of 60x60 km^2, as shown in Figure 6.7.

Primary TV transmitters are assumed to be located at well-known positions outside the CRN perimeter and far away from the BS, at a maximum distance of 150 km. The area is divided into a grid of 256 square cells of $\frac{150000}{\sqrt{256}} = 9375$ meters side. As explained in Section 6.2.3, we assume that the BS stores a database with the TDoA measures that each anchor node should report if transmission sources were located at the center of every cell.

Since TDoA measures are subjected to errors which depend on the SNR (see Section 6.2.3), we set a fixed value for the SNR at the BS, i.e., SNR_0. Then, the SNR values for each anchor node are computed with respect to SNR_0 according to the path loss model given by (6.63).

We have also considered a malicious user performing a PUE attack from different positions both inside and outside the CRN area.

6.3.3 Evaluation of the location error

In this section, we analyze the location error performed with the proposed mechanism using the methodology explained in Section 6.3.1. We provide the simulation results obtained with Matlab for a wide variety of scenarios, aiming at studying the dependence of the location error on the following parameters:

- *Location of the attacker.* The impact of a PUE attack will vary depending on the location of the attacker. Because of this, we have run simulations considering four representative positions: an attacker lying inside the CRN area, at the border of the CRN. outside the CRN at a medium distance between a real primary user and the CRN, and pretty close to a real primary user.
- *Number of cooperating CRs.* As mentioned in Section 6.1.1, cooperation among different nodes can increase the accuracy of location. In this work, we have considered a number of cooperating CRs ranging from $n = 10$ to $n = 100$.
- Average *SNR* at reception. The accuracy of TDoA-based location methods strongly depends on the received SNR. Due to effects such as shadowing or fading, low SNRs are expected when receiving primary signals [Hoven and Sahai, 2005]. As a consequence, we have run simulations considering an average SNR at reception -10dB, 0dB and 10dB.

Case 1: PUE attacker inside the CRN

Figures 6.11, 6.12 and 6.13 depict the results obtained considering an emitter inside the CRN area at position $(x, y) = (8000, 1000)$, for a SNR at reception of -10, 0 and 10 dB respectively. The ellipses show the 99% confidence interval error computed over 10000 simulations. Under the worst conditions, i.e., SNR=-10dB, the error ranges from approximately 30 meters when $n = 100$ stations are cooperating, to around 491 meters in the major axis for $n = 10$. In any case, these errors are acceptable when dealing with a network deployed over an area of 60-by-60 km^2. Besides, the emitter can be located with enough precision so as to distinguish it from a real primary user. It can also be observed that, as the SNR increases, the error is gradually reduced. For SNR=10dB, it ranges from a value around 40 meters for $n = 10$ to 3.3 meters for $n = 100$ stations. Note that, besides the SNR value, the number of cooperating nodes has a big impact on the performance of the mechanism, so there is a tradeoff between the amount of resources needed for the location process and the efficiency of the mechanism.

However, above 40 nodes the accuracy of the estimation hardly improves in comparison with the increase of the number of stations cooperating. As a result, $n = 40$ seems to be the optimal value under such conditions.

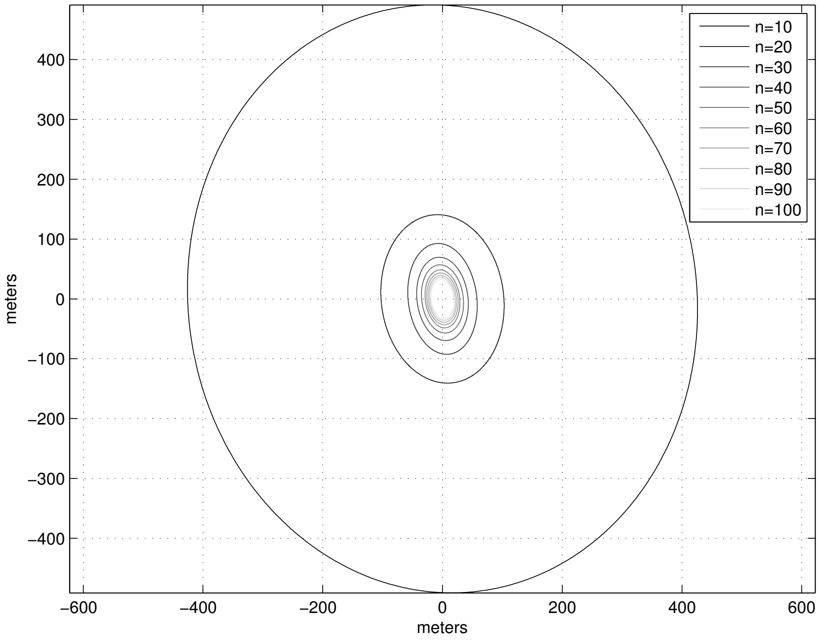


Figure 6.11: 99% confidence interval error (10000 experiments), (x,y)=(8000,1000), SNR=-10dB

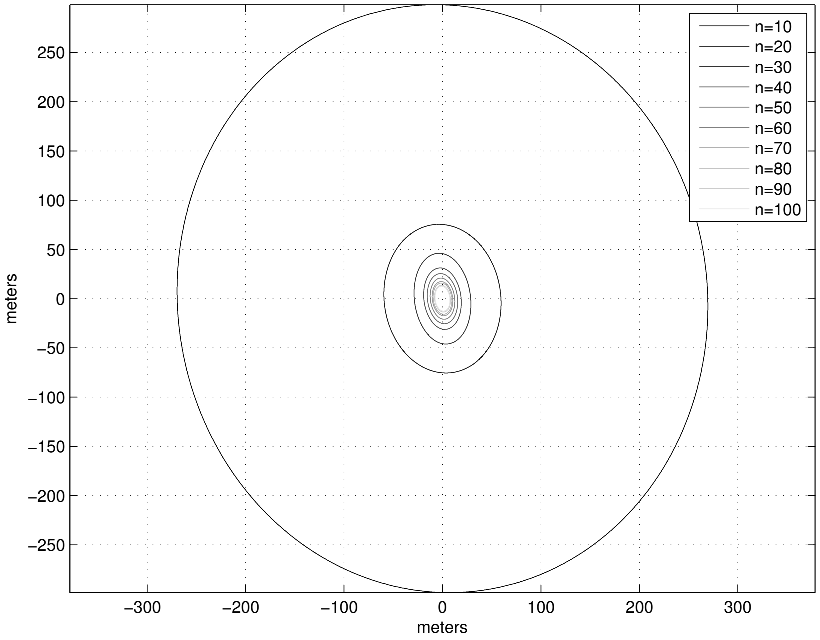


Figure 6.12: 99% confidence interval error (10000 experiments), (x,y)=(8000,1000), SNR=0dB

On the other hand, as the variance of the TDoA measures is inversely proportional to the linear value of the SNR (6.62), the estimation error increases as the SNR decreases proportionally to the square of the variance. In our simulations, we have considered SNR$_1$=10dB, SNR$_2$=0dB and SNR$_3$=-10dB. As a consequence, the linear relation between the respective variances will be as in (6.71):

$$\sigma_2^2 = 10\sigma_1^2 \qquad \sigma_3^2 = 10\sigma_2^2 = 100\sigma_1^2 \tag{6.71}$$

This implies that, for a given number of cooperating nodes, when the SNR is -10dB the error will be approximately $\sqrt{10}$ times bigger than when it is 0dB, and ten times bigger than when it is 10dB.

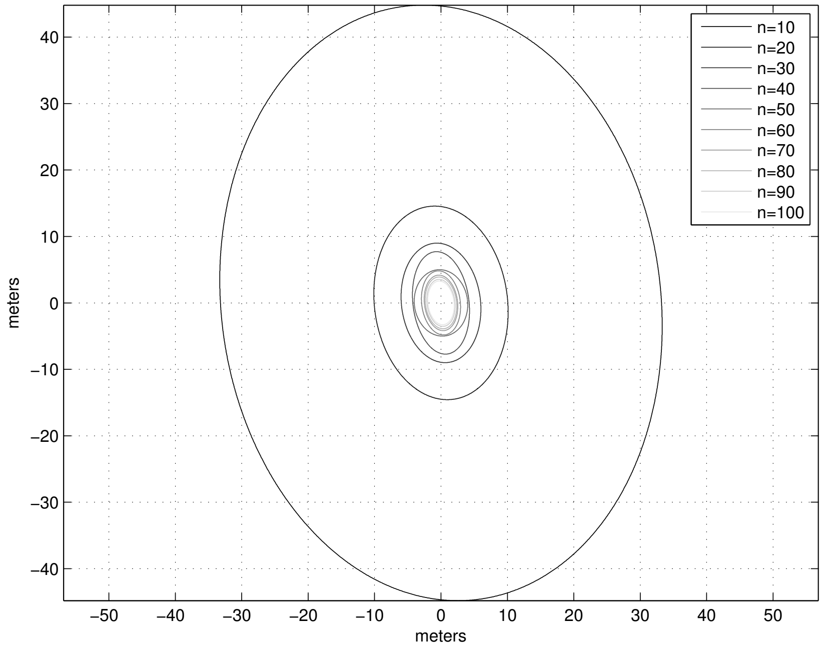


Figure 6.13: 99% confidence interval error (10000 experiments), (x,y)=(8000,1000), SNR=10dB

Case 2: PUE attacker at the border of the CRN

In this section, we consider a PUE attacker lying at the border of the CRN at position $(0, 30000)$. This implies that range distances d_i, i.e., the distance between the emitter and node i, will be considerably large for most CRs. Figures 6.14, 6.15 and 6.16 depict the 99% confidence interval of the location error, again for SNRs at reception of -10, 0 and 10 dB respectively. For a SNR=-10dB, the distance error at the major and minor axis range from 743.4 and 280.5 meters, when $n = 10$ stations are cooperating, to around 24.3 and 15.56 meters for $n = 100$. As in the previous case, the location error is further decreased as improves the SNR at reception. For SNR=10dB, the distance error at the major and minor axis are of 97.16

and 31.66 meters when $n = 10$ stations are cooperating. For $n = 30$, the position estimate is already very precise, with errors of 8.19 and 5.124 meters in each axis.

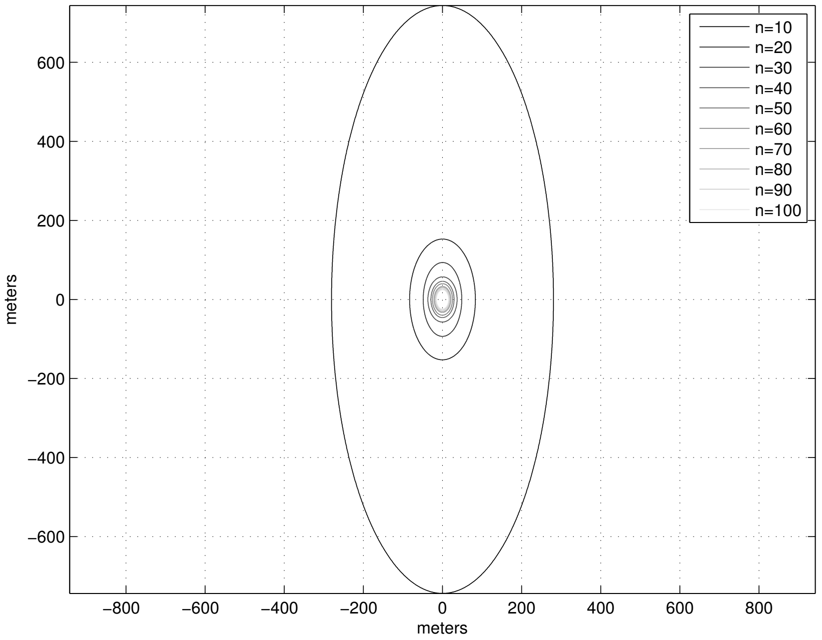


Figure 6.14: 99% confidence interval error (10000 experiments), (x,y)=(0,30000), SNR=-10dB

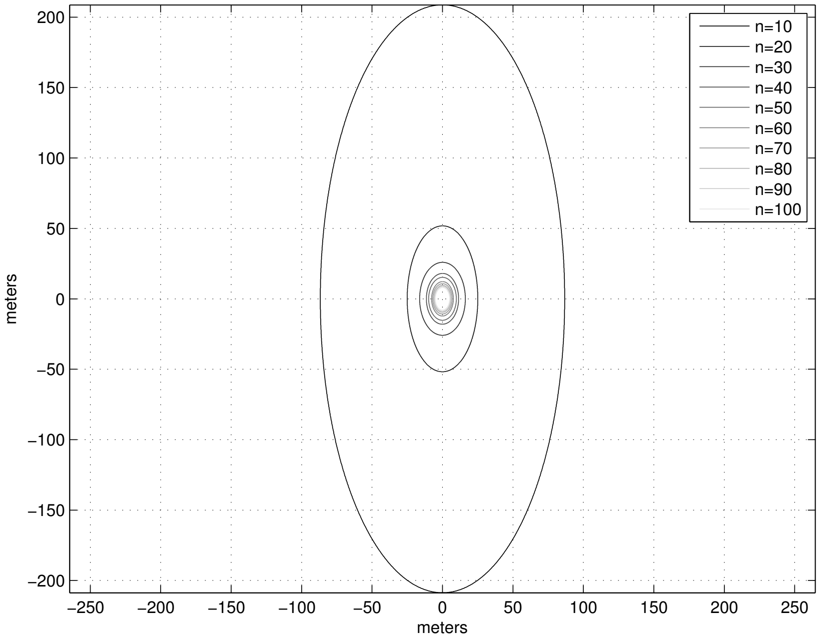


Figure 6.15: 99% confidence interval error (10000 experiments), (x,y)=(0,30000), SNR=0dB

It can also be observed that the location error is always much greater in the ordinate axis than in the abscisae axis. This is due to the position selected by the attacker $(0, 30000)$. From

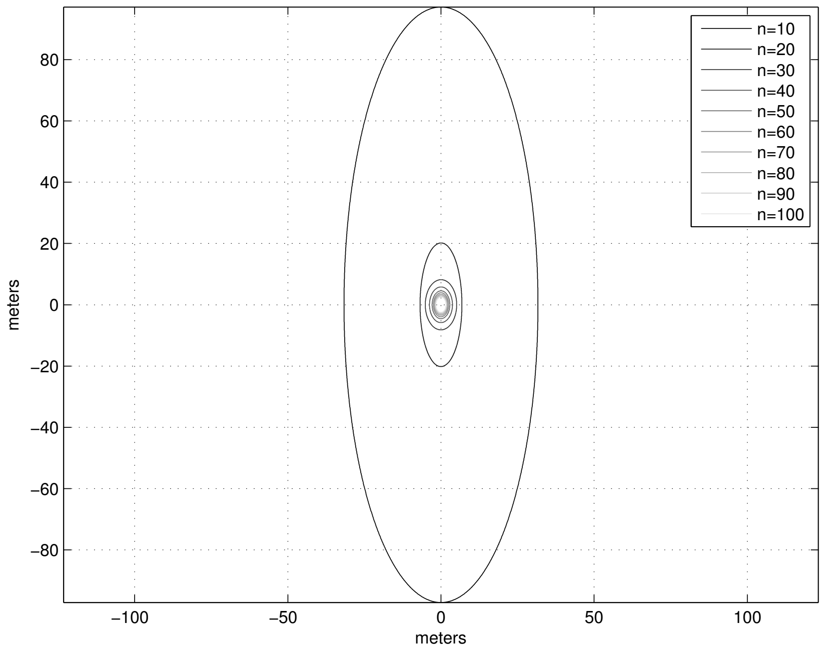


Figure 6.16: 99% confidence interval error (10000 experiments), (x,y)=(0,30000), SNR=10dB

this observation, one can infer that the most far away a given coordinate is from the BS, the bigger the location error in that coordinate. On the other hand, the location error increases proportionally to the square root of the linear increase in the SNR, as in the previous case.

Case 3: PUE attacker outside the CRN

When the attacker has enough resources regarding transmission power, it can perform the PUE attack from a position outside the CRN. Figures 6.17 and 6.18 show the results obtained considering now an emitter at position $(x, y) = (-75000, 50000)$, for a SNR at reception of -10 and 10 dB respectively. As the coordinate differences between the BS and the attacker are further increased, the location error is much higher than in those simulations where the emitter lies inside the CRN. In the worst case (SNR=-10dB), the error ranges from values around 1.39 km in the major axis for a small number of cooperating nodes, i.e., $n = 10$, to 137m for $n = 100$. Assuming that primary emitters are located around 150 kilometers far away from the BS, as it is required in the 802.22 standard, the accuracy provided by the method is still high enough in order to avoid false negatives, i.e., an emitter could located at this position would be considered a PUE attack

We have omitted the simulation results for SNR= 0dB, since they provide no additional information. As previously observed, there is a direct relation between the location error and the SNR and therefore these results can be approximate from the values obtained for SNR= 10dB or SNR=-10dB.

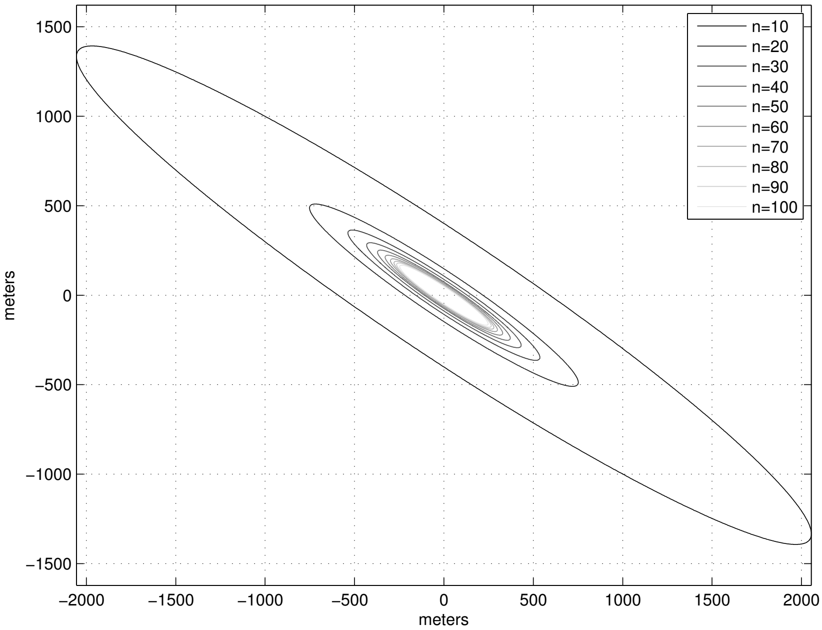


Figure 6.17: 99% confidence interval error (10000 experiments), (x,y)=(-75000,50000), SNR=-10dB

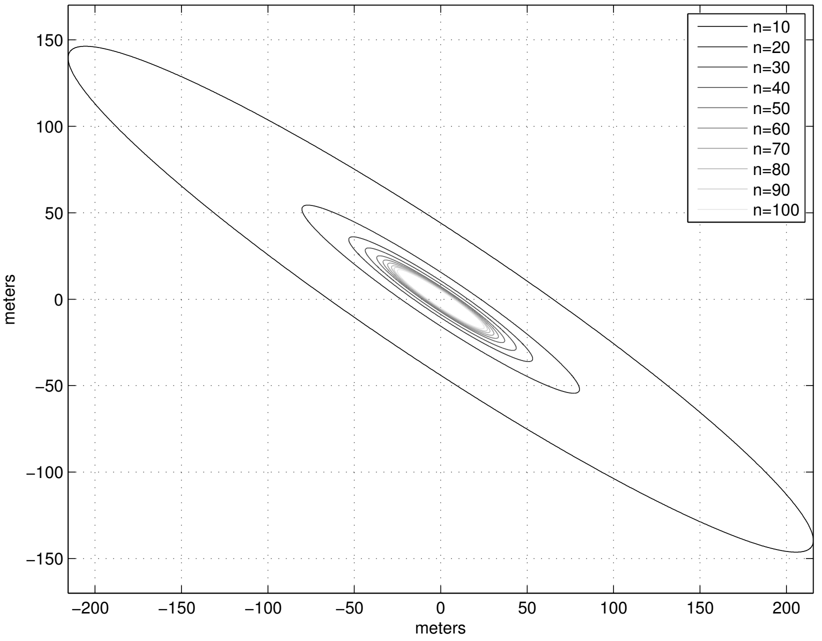


Figure 6.18: 99% confidence interval error (10000 experiments), (x,y)=(-75000,50000), SNR=10dB

Case 4: PUE attacker near a primary user

In this last scenario we have considered a malicious user performing a PUE attack in the vicinity of a true primary user, at position $(140000, 0)$. Although it represents an unlikely situation because it would imply an attacker with power transmission capabilities similar to those of real TV towers, the possibility must be considered. Figure 6.19 depicts the 99% confidence interval of the location error for an average SNR at reception of -10dB. In this case, we have omitted the simulation results for other values of SNR because of two reasons: 1) as mentioned above, the values can be approximately obtained by using the relation in (6.71), and 2) it is unlikely to expect higher values of SNR at reception for such long distances, since this would imply an attacker transmitting at a higher power than TV stations, which could cause interferences to TV transmissions.

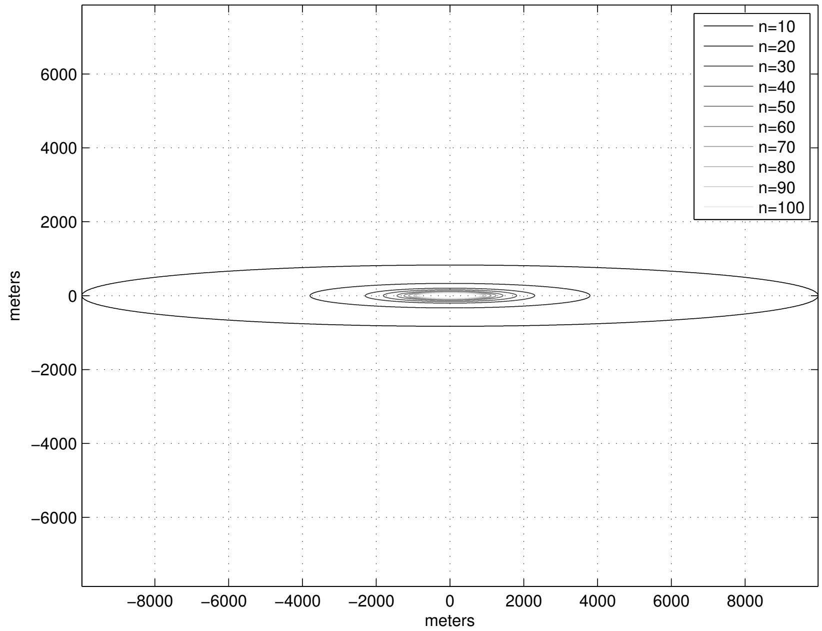


Figure 6.19: 99% confidence interval error (10000 experiments), (x,y)=(140000,0), SNR=-10dB

It can be observed that distance errors are now on the order of kilometers: with 9978 and 829.1 meters for the major and minor axis when $n = 10$ to around 841.3 and 3.46 meters for $n = 100$. These results clearly stress the risk of a successful PUE attack when the attacker lies near a real primary user. As in case 2, the location error is always much greater in one of the axes (x) because of the position of the attacker. This suggests that, in order to decrease the accuracy of the location process of the CRN and increase the probability of a successful PUE attack, both the attacker and the true primary user should be as far as possible from the BS with respect to both coordinates x and y.

Discussion

Based on these results, we can conclude that in most of the cases the accuracy of the proposed location method is enough to effectively distinguish between a true primary transmission and a PUE attack. An attacker could strategically select a position inside the CRN and perform power control leading to false primary detection when only RSS measures are considered. In such case, however, the positioning method would allow to unequivocally locate the emitter with errors of at most hundred of meters, far away from real primary users.

A PUE attack can be more effective if it is performed from a point located outside the CRN, in particular when the position of the attacker is pretty close to a primary user. Assuming that a large number of CRs are cooperating in the location process, the attacker should be at most 1 kilometer far away from the primary transmitter. However, in this case the attacker would need more resources to transmit a signal with enough power to be detected within the CRN. This would not be the case of a typical CRN attacker but rather that of a pirate TV tower transmitting in the vicinity of a legitimate primary user. Because it would represent a higher threat to the primary network than to the CRN, it is out of the scope of the CRN prevention mechanisms.

Detection of a PUE attack must lead to policy-defined counteract actions ranging from reporting the attack to the "police" to ignoring the fake primary signal and continue interfering it. Besides effective detection of PUE attacker, a CRN must ensure protection to primary users. False positives in the location method, i.e., to consider a legitimate emitter as an attacker, could lead to interfere with primary transmissions, which is strictly forbidden by CRN regulations [FCC, 2003b]. According to this, minimizing the probability of false positives becomes of paramount importance. In the case of our proposal the probability of false positive can be expressed as $1 - \alpha$, with α the confidence level. In order to satisfy the requirement of the 802.22 standard, which establishes a maximum probability of miss detection of primary users of 0.1, the location method should be properly designed. In particular, it should guarantee that under the worst conditions, i.e., SNR=-10dB, the probability of confusing a legitimate primary user with a PUE attacker is less than 0.1. This goal can be achieved in two ways. On the one hand, by requiring the cooperation of a large set of CRs, i.e., at least $n = 100$ stations, in the location process. On the other hand, if such amount of stations are not available, this probability can be minimized by performing the location process m times. This leads to a probability of false positive of $P_f = (1 - \alpha)^m$. Therefore, if a given P_f is desired, at least $m_{min} = \lceil \log_{1-\alpha} P_f \rceil$ iterations should be carried out. For example, if we want a probability of false positive of 10^{-6} with a confidence level of 99%, 3 different iterations will be needed.

6.3.4 Evaluation of the location process time

In this section we evaluate the time needed to locate an given transmission and in particular, the overhead introduced by the location method, i.e., the amount of time needed by the CRN

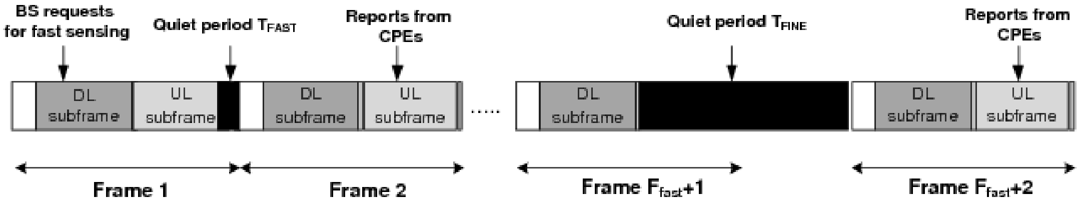


Figure 6.20: IEEE 802.22 sensing process

anchor nodes to transmit the recordings to the BS.

Figure 6.20 depicts the process of detecting a primary signal. The first step is to detect a potential primary transmission by performing fast spectrum sensing. Upon request of the BS, each node takes an energy measure and reports it to the BS in the next frame, within the slot assigned by the BS. The BS collects all reports and takes a global decision about the existence of a primary user. Because measures are subjected to errors, the BS may not detect the primary transmission at a given iteration but in one of the subsequent ones, needing F_{fast} frames for fast detection.

Let p be the probability of detection of the mechanism used by the BS, then the average number of sensing iterations $\overline{N}_{FAST}$ needed in order to detect a primary signal can be computed as in (6.72).

$$\overline{N}_{FAST} = \sum_{k=1}^{\infty} k(1-p)^{k-1}p = \frac{1}{p} \tag{6.72}$$

As a consequence, the average amount of time $\overline{T}_{FAST}$ needed can be computed as in (6.73),

$$\overline{T}_{FAST} = \frac{1}{p}\, 2\, T_{FRAME} \tag{6.73}$$

where T_{FRAME} is the time length of the 802.22 frame.

After fast sensing is performed, the BS may schedule a longer quiet period to perform fine sensing in the next frame $F_{fast}+1$. This period could also be used by the BS to send a marker signal and to request the set of nodes to make the recordings, in order to obtain the set of TDoA measures. The BS requests are sent in the DL subframe of frame $F_{fast}+1$. The quiet period required to perform fine sensing and make the recordings may start at the beginning of the UL subframe, also in frame $F_{fast}+1$. Therefore, the reports containing the sensing fine measures and the recordings can be sent in the UL subframes of subsequent frames, i.e., in frames $F_{fast}+2$,$F_{fast}+3$, etc. The measures regarding fine sensing can be transmitted in a single frame [802.22 WG, 2011]. In the following, we derive the number of extra frames needed to send the recordings.

In order to correlate the recording made by a given node and the BS, each recording should be at least as long as the maximum propagation delay within the network. Considering a network with a radius of 30 Km, the maximum propagation delay will be 100μs. On the other hand, in the ATSC system the length of a DTV frame is 77 μs. Therefore, in order to satisfy the requirement mentioned above and to be able to decode at least one DTV frame, we have considered an amount of recording time of $2 \cdot 77\mu$s$= 144\mu$s.

Recall the structure of the OFDMA frame structure shown in Figure 1.5. Each OFDMA symbol is divided into subchannels of 28 sub-carriers. Out of the 28 sub-carriers, 4 are pilots and therefore it leaves 24 subcarriers per subchannel for data transmission. As a result, the number of data bits per subchannel and per OFDMA symbol is $24 \cdot \log_2 M \cdot CR$, with M the number of symbols of the modulation and CR the coding rate. There are a total amount of 60 subchannels; 2 are reserved for special purposes such as ranging or bandwidth request and 58 are available for data transmission.

Let us denote L_{record}, the amount of bits to be recorded by every anchor node; S as in (6.74) the number of slots that should be assigned to every anchor node in order to transmit the recorded data; and S_{ul} the number of symbols per subchannel assigned to the UL subframe. As a consequence, the number of frames F_l needed to transmit the recordings from n anchor nodes can be computed as in (6.75).

$$S = \left\lceil \frac{L_{record}}{24 \cdot \log_2 M \cdot CR} \right\rceil \tag{6.74}$$

$$F_l = \left\lceil n \cdot \frac{S}{58 \cdot S_{ul}} \right\rceil \tag{6.75}$$

Considering the worst case a maximum data rate of 19.4 Mbps in DTV, the amount of data to be recorded by each anchor node is $L_{record} = 144\mu$s$\cdot$19.4 Mbps= 3348bits. Table 6.1 shows the number of frames and the corresponding time needed for transmitting the anchors' recordings to the BS when the number of cooperating stations is $n = 10$ and $n = 100$. We have chosen representative values for modulation and coding rate. The number of symbols per subchannel depend on the bandwidth of the TV channel and on the CP (see Section (1.1.4)). In 6MHz bandwidth channels, the number of symbols per subchannel ranges from 26, for CP=$\frac{1}{4}$), to 31 symbols for CP=$\frac{1}{32}$). We have considered two different values $S_{ul} = 10$ and $S_{ul} = 20$, for the number of symbols reserved for the UL subframe, i.e., that can be used to send the recordings.

Table 6.1 shows, as expected, that the less cooperating anchor nodes the less additional delay introduced by the location process. It can be observed that the minimum amount of time needed for transmitting the anchors' recordings to the BS is one frame, i.e., 10 ms, for 64-QAM $\frac{5}{6}$ modulation and $n = 10$ anchor nodes. However, in order to obtain accurate estimations of the emitter's position, a large number of cooperating anchor nodes is needed. Even in this case, the time needed to transmit the recordings is limited to a few hundred of milliseconds. In the worst case, i.e., QPSK $\frac{1}{2}$ modulatin, $n = 100$ anchor nodes and $S_{ul} = 10$, the time needed rises to 25 frames, i.e., 250 ms. These results prove the feasibility of performing the location

Mod. (CR)	S_{ul}	F_l (time) $n = 10$	$n = 100$
QPSK $\frac{1}{2}$	10	3 (30ms)	25 (250ms)
	20	2 (20ms)	13 (130ms)
16-QAM $\frac{2}{3}$	10	1 (10ms)	10 (100ms)
	20	1 (10ms)	3 (30ms)
64-QAM $\frac{5}{6}$	10	1 (10ms)	5 (50ms)
	20	1 (10ms)	3 (30ms)

Table 6.1: Frames/time for localization process

process in a reasonable amount of time without diminishing the capacity of the network.

6.4 Conclusions

CRNs rely on spectrum sensing mechanisms to identify white spaces in the spectrum left unused by primary users. A malicious user can take advantage of this feature by emulating a primary transmission, i.e., perform a PUE attack, and prevent the CRN from using available bands.

In this chapter we have proposed a cooperative method to effectively detect PUE attacks based on TDoA location. The method can be applied to infrastructure-based networks where the location of true primary users is known, as it is the case of TV emitters in WRAN 802.22 networks. The rationale behind it is to estimate the location of the emitter and compare it with the known positions of primary users so as to identify potential attacks.

During the location process, CRs are requested to record a fragment of the received signal and send it to the BS, which derives a set of TDoA measures by correlating the signal received by itself with those coming from the CRs. The non-linear system of equations given by this measures is then solved by applying Taylor series estimation, which applies a WLS criterion in order to obtain a position estimate while minimizing the sum of squares error. When applying a WLS method, estimation accuracy can be improved with respect to the basic LS mechanism by assigning more weight to those measures that are expected to be more reliable.

The method poses several challenges that we have addressed as follows. First, TDoA measurement require a tight synchronization between the BS and the CRs. In order to overcome this problem, the BS transmits a marker signal which is recorded by the CRNs together with the emitter's signal, and allows to synchronize every pair of recordings. Second, Taylor series estimation requires an initial guess of the emitter position and the convergence of the algorithm strongly depends on it. In order to obtain this initial guess, we have proposed to use a database with pre-stored TDoA measures.

Our proposal has been evaluated via simulation under a set of scenarios with different environment conditions and different positions for the attacker. The goodness of the method has been proved based on two different metrics: location accuracy and duration of the location process. The simulation results show that an attacker can be easily identified whenever the PUE attacker is located inside the CRN or at a position relatively far away from true primary users. However, as the bigger the distance between the emitter and the CRN, the lower the location accuracy, the optimal attack can be performed by selecting a position near a primary user. In such case, the location process exhibits errors of the order of kilometers, increasing the probability of a successful PUE attack. Despite it, the optimal attack requires a big amount of resources with regard to power transmission and this probably wouldn't be available to most attackers. Besides, such attack would probably interfere primary transmissions, pushing the responsibility of detecting it and providing the corresponding countermeasures to the primary network. On the other hand, the amount of time required to perform the location process has been shown to be bounded to reasonable values, meaning that the overhead introduced by the method does not represent a significant decrease of the CRN capacity.

We can conclude that the proposed method can effectively deal with PUE attacks and prevent CRNs from losing spectrum opportunities. However, further work is needed in order to refine the method and improve its performance. On the one hand, position accuracy could be improved by applying hybrid-mechanisms which rely on different types of measures. Given that the inherent spectrum sensing mechanisms of CRNs already require RSS measures, an interesting approach would be to evaluate the performance of location methods by combining RSS and TDoA measures. On the other hand, a more realistic model where the effect of multipath is taken into account should be evaluated. Multipath is one of the major sources of TDoA measures errors, but it is also difficult to model because it depends on the particular characteristics of the terrain where the network has been deployed.

Chapter 7

Securing Cooperative Localization for PUE Attacks Detection

Several mechanisms, such as spectrum sensing or the location method to detect PUEs attacks proposed in Chapter 6, are based on the cooperation of the CRs associated to the network. These nodes are required to provide with information to a fusion entity, which combines all data and applies a particular algorithm in order to take a decision, e.g., a primary exists or a PUE attack is being performed.

These cooperative mechanisms rely on the assumption that CRs are willing to cooperate and, even more important, that these nodes are honest and their measures may be deviated with respect to the true value only due to degradations of the channel. However, such mechanisms can be easily compromised either by outside attackers trying to disrupt the normal operation of the CRN, selfish nodes which do not report any information in order to save resources or malicious nodes reporting false feedback.

An outside attacker can jam the communication channel used by the CRN, thus preventing cooperating nodes from providing with the measurements needed for the cooperative algorithm. The attacker can be motivated to perform the attack, for example, because it can benefit from the spectrum left available by the CRN (due to the failure of the cooperative algorithm in detecting a vacant portion of the spectrum) or just because the attacker is trying to degrade the performance of the network.

Selfish behavior reduces the amount of available measures for the cooperative method and can become a big risk as the percent of selfish nodes increases. By way of illustration, the location method proposed in Chapter 6 achieves higher accuracy in the position estimation as the number of range measurements becomes higher. Nevertheless, such undesired behavior can be prevented in CRNs with cooperation enforcement mechanisms [Marias et al., 2006] without altering the cooperative method. For example, in 802.22 networks where the BS is responsible for distributing the available bandwidth among the CRs, it may reduce or even cancel the bandwidth assigned to a given node if it does not collaborate.

Finally, the effect of malicious CRs providing false data can severely undermine the cooperative method and can lead the CRN to take wrong decisions. In this chapter, we approach the problem of malicious CRs reporting false measurements to a BS in charge of performing the location method described in Chapter 6. As previously explained, this method is based on TDoA measures and is targeted to locate the source transmission in order to distinguish between legitimate primary transmissions and PUE attacks. Because a successful PUE attack can considerably decrease the performance of a CRN, it is essential to make the location method robust to false feedback. The proposed approach is related to the implementation of the reliability module presented in Chapter 4, which should be embedded into an attack detection system for CRNs, as shown in Figure 7.1.

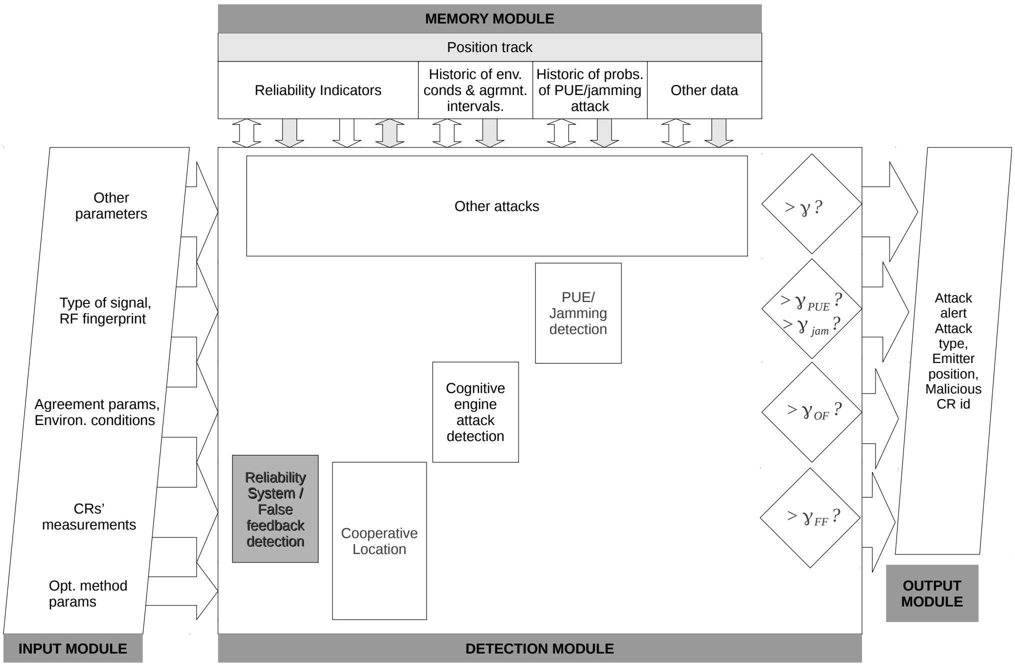

MEMORY MODULE
Position track
Reliability Indicators
Historic of env. conds & agrmnt. intervals.
Historic of probs. of PUE/jamming attack
Other data
Other parameters
Type of signal, RF fingerprint
Agreement params, Environ. conditions
CRs' measurements
Opt. method params
INPUT MODULE
Other attacks
PUE/ Jamming detection
Cognitive engine attack detection
Reliability System / False feedback detection
Cooperative Location
> γ ?
> γ_PUE ?
> γ_jam ?
> γ_OF ?
> γ_FF ?
DETECTION MODULE
Attack alert Attack type, Emitter position, Malicious CR id
OUTPUT MODULE

Figure 7.1: Reliability module of a security system for CRNs

The structure of the chapter is as follows. Section 7.1 gives an overview of the proposals appeared in the literature dealing with false feedback. In Section 7.2, we define the attacker model and analyze the effect of false measures on the location method. A method for detecting and counteracting such attacks is proposed in Section 7.3, and evaluated through simulation. Finally, in Section 7.4 we provide the conclusions of this work.

7.1 Related work

Detection of false feedback in cooperative methods for CRNs has been object of study in previous work, mainly with regard to spectrum sensing for primary user detection [Chen et al., 2008a, Chen et al., 2009a, Qin et al., 2009, Min et al., 2010, Kaligineedi et al., 2008]. The methods presented in [Chen et al., 2008a, Chen et al., 2009a, Qin et al., 2009] rely on hypothesis tests where the sensing measurements provided by the set of CRs are weighted according to their reputation. The particular reputation value for a node is updated based on the correlation between the measure reported by the node and the global decision about the existence of a primary user taken by the CRN fusion entity. This fact implies that a global decision is first needed in order to detect false reports and compute the reputation value. Analogously, the location method would require an initial estimation of the position so as to identify abnormal measurements and detect a FF attack. As it will be seen in Section 7.2, the location method is very sensitive to false data and a single forged measurement could lead to a completely erroneous estimation. Because of this, these methods cannot be applied directly to secure the location process.

Other methods, such as [Min et al., 2010], rely on the correlation among measurements from neighboring nodes in order to detect abnormal values. This approach achieves reasonable performance for small and dense networks but may be useless in large networks, where nodes are distanced more than a hundred of meters and correlation among their measurements may almost be null.

In [Kaligineedi et al., 2010] different methods based on outliers detection techniques [Kaligineedi et al., 2008] are analyzed. These techniques rely on different parameters such as the mean, the standard deviation or the median, in order to identify those measurements which clearly deviated from the data set.

On the other hand, the design of location methods resistant to FF attacks has also been addressed by the research community in the context of wireless sensor networks [Srinivasan and Wu, 2007]. Many applications developed for such type of networks require sensor nodes to estimate their location. Typically, the protocols used for location discovery are based on the use of special nodes called beacon nodes, which are assumed to known their own locations. These nodes transmit special radio signals called beacon signals which carry packets with information about the location of the beacon node. Based on these beacons, non-beacon nodes can estimate their position applying any of the range methods described in Section 6.1. However, a sensor node wishing to obtain its position must face the problem, among others, of malicious nodes providing incorrect location references.

The location method we have proposed for CRNs differs with respect to the location approach in wireless sensor networks fundamentally in three aspects. First, the location target is an attacker rather than a member of the network. Second, the entity in charge of performing the location process is the BS instead of the node itself. Finally, wireless sensor networks typically have an ad hoc structure, and the data sent by a given node by means of beacon signals can be contrasted by other neighboring nodes. In CRNs such as 802.22 networks, only the BS is typically responsible for verifying data. Despite it, both approaches follow the same principle in the sense that the position of a given node is estimated by means of measurements performed by anchor nodes with known positions. For this reason, existing countermeasures in this area should be also considered for securing the location process in CRNs.

In [Liu et al., 2005], two different approaches are proposed in order to counteract false feedback. The first method detects false references according to their inconsistency with regard to honest ones. First, the node estimates its location by means of the reference distances provided by the set of beacons and applying a minimum mean squares method. Based on such estimation, the inconsistency of every reference is measured by computing the mean square error. The second approach is based on a voting process. The target field is divided into a set of cells and the node estimates the most likely cell according to the different references. The most voted cells are selected, and the position estimation is computed as the center of these cells.

The authors in [Li et al., 2005] propose the use of a Least Median Squares (LMS) approach, an improvement to the LS estimation which allows to identify deviated references.

7.2 False feedback attacks to the localization method

Prior to the design of an effective countermeasure, there is a need for characterizing the attacker and evaluating the impact of false feedback on the location process. In this section, we define the behavior of a malicious node trying to disrupt the location method by forging TDoA measures and, based on this definition, we provide the simulation results obtained when location is performed under the presence of such undesired behavior.

7.2.1 Definitions and attacker model

We define a liar as any node announcing false data to the BS in charge of applying the TDoA localization method. Liar nodes are assumed to be authenticated in the network and know the cryptographic key used for communication with the BS, if used. As a consequence, the BS will initially trust false data provided by such nodes. The intent is assumed to be malicious with the purpose of misleading the BS into a wrong calculation of the emitter's position.

Figure 7.2 depicts the FF attack. We have considered that the TDoA measure τ_{i0}^{f}, obtained by the BS because of the false data reported by node i is as in (7.1), with d_{i0} the distance between the liar and the BS, d_{i2} and d_{02} the false distances from the emitter to the node i and the BS respectively, v_p the propagation speed and ε_i the error performed in the measure following a normal distribution with zero mean and variance as in (6.62).

$$\tau_{i0}^{f} = \frac{d_i^1 - d_0^1}{v_p} = \frac{d_{i0}}{v_p} + \varepsilon_i \tag{7.1}$$

According to the position of the emitter, the true TDoA measure τ_{i0}^{t} obtained from the pair BS and node i should be as in (7.2).

$$\tau_{i0}^{t} = \frac{d_i^2 - d_0^2}{v_p} + \varepsilon_i \tag{7.2}$$

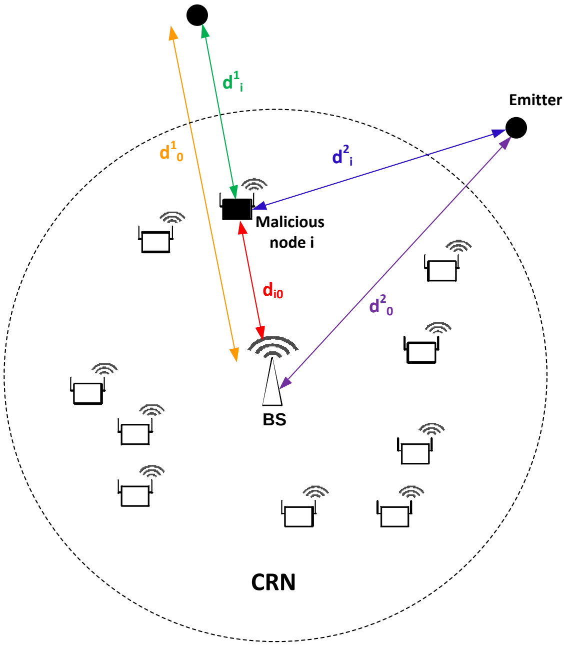


Figure 7.2: False measure reported by a malicious node

The wrong TDoA measure computed because of the attack represents the maximum TDoA value that can be obtained by means of the pair node i and BS. Measures above this value would be discarded by any BS because they are not possible at all. In practice, such measure can only be obtained when the emitter, node i and the BS are aligned, as shown in Figure 7.2. This implies that, in general, the false TDoA measure will always have a higher absolute value than the real one. Thus, the forged report will contribute to deviate the position estimation from the real position of the emitter. Note that, because a single TDoA measure cannot force the location method to get an specific wrong estimation, the most possible damage into the algorithm can be achieved by forging the measure in this way.

7.2.2 Effect of false measures on the position estimation

In order to analyze the impact of liars on the location process, we have run a set of simulations considering a 802.22 network deployed over an square area of 60x60 km^2. The network is composed by a BS located at the origin (0,0) and a set of n cooperating CRs ranging from 10 to 100, uniformly distributed within the area of the CRN. We have considered a malicious user performing a PUE attack positioned at (x, y), where x and y are given in meters. The average SNR at reception in the BS is -10dB. The Okumura-Hata model (6.64) has been adopted as path loss model and used to derive the reception SNR at the set of CRs. For the location process, the method described in Chapter 6 is applied in order to obtain the position estimation. Besides, we have considered the existence of a varying number of liars (l out of the n cooperating nodes) ranging from 1 to 3, providing false measures to the BS, as it has been explained in the previous section.

Impact of the number of liars

Aiming to analyze the impact of the number of malicious nodes reporting false feedback, we have considered a PUE attacker placed at a fixed position (30000,0) and a varying number of liars. Table 7.1 provides the 99% confidence interval error of the position estimation over the 10000 simulations. In each simulation, the error has been obtained by computing the Euclidean distance between the real position of the emitter and the position estimated by the location method. The simulation results show that the presence of just one liar results in a very inaccurate prediction of the position, with errors rising from order of meters to kilometers, for any number of cooperating nodes. In particular, for a number of nodes below 40, the FF attack can easily lead to a false positive regarding the existence of a primary user. Although for higher values of n the accuracy is also very low, the risk of false positive is also lower because primary users are expected to be around 150 km far away from the BS. However, the attack could success if the PUE attacker was positioned nearer to a real primary user regardless of the number of cooperating nodes. As seen in Chapter 6, the location accuracy decreases as the distance between the emitter and the anchor nodes increases. Moreover, it is required higher precision in the location method when the emitter is placed near a primary user, in order to effectively distinguish an attacker from a legitimate transmitter.

On the other hand, note that, as the number of liars increases, the distance error becomes higher but its increase is not proportional to the number of liars. This is because malicious nodes can forge their measurements and disrupt the location process but have no control on the effect of the provided false data on the final estimation of the position. As we will see later on, the way in which the position estimation is shifted (with regard to the true value) due to a forged measure depends on the position of the malicious node. As we are assuming that CRs are static and randomly positioned over the CRN area, they cannot force the location method

n	no liars	1 liar	2 liars	3 liars
10	358.9	112000	117960	120240
20	75.04	66779	98175	100580
30	46.64	34105	53794	70607
40	35.28	20948	34933	45971
50	30.06	12642	22165	30316
60	26.12	9626	15359	19942
70	23.4	8087	11094	14452
80	21.24	6562	9038.6	11699
90	19.85	5078	7631	9900
100	18.82	4442	6066	7509

Table 7.1: 99% confidence interval error of the position estimation (meters)

to derive an specific wrong estimation.

Figure 7.3 and 7.4, depict the prediction error for the 10000 experiments when 30 nodes are cooperating and there are no liars and one liar, respectively. As mentioned before, introducing false data into the location method leads to a devastating damage in terms of location accuracy. Besides, it can be observed that under the presence of liars most error samples are clearly deviated to the positive abscissa axis and adopt an arrow shape. This is because the emitter lies on the positive axis and, when a forged measure is introduced into the location method, the position which minimizes the sum of square errors (given by the WLS algorithm) lies most of

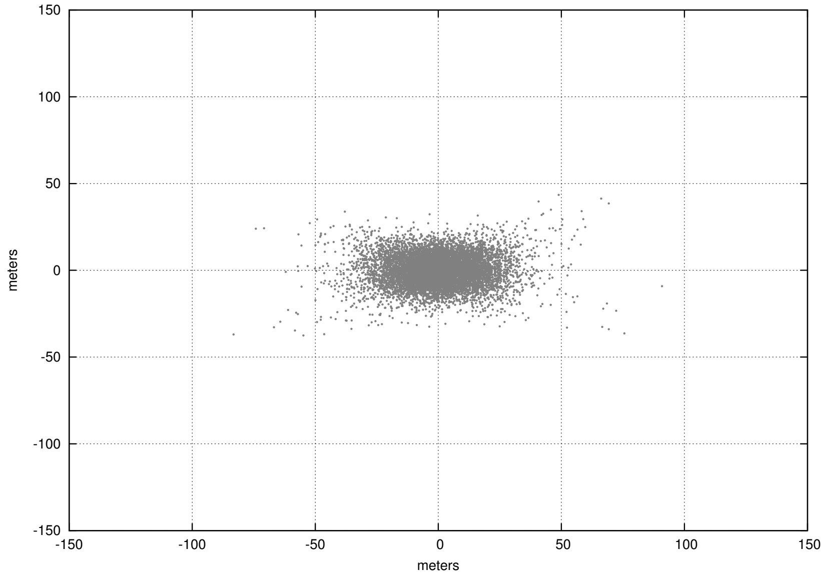


Figure 7.3: Prediction error with no liars (10000 experiments),(x,y)=(30000,0),n=30

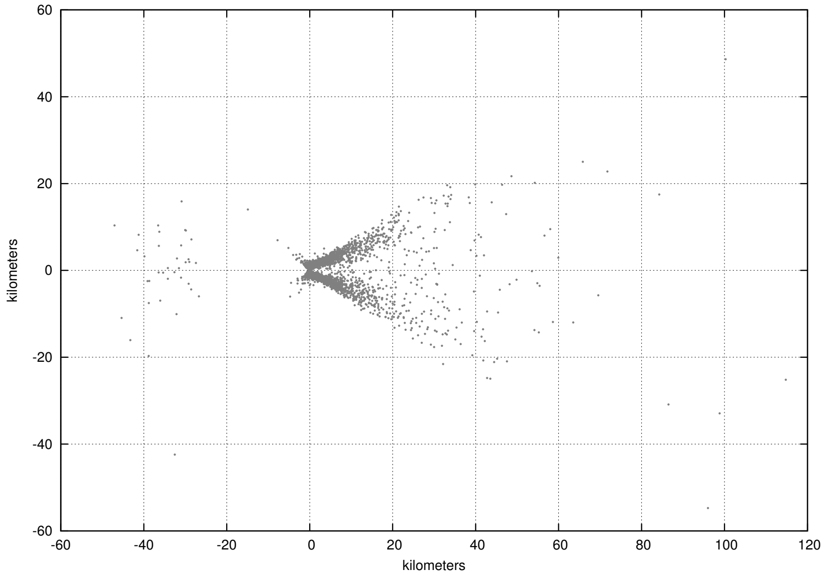


Figure 7.4: Prediction error with 1 liar (10000 experiments),(x,y)=(30000,0),n=30

the times on a positive abscissa coordinate. As most of the nodes are reporting true measures, a position estimation lying on the negative abscissa axis would lead to higher values of the sum of squares error. It can also be seen that samples are symmetrically distributed between the positive and the negative ordinate axes. This effect is given by the location of the liar node, as shown in Figure 7.5 : when the malicious node is located at a positive ordinate, the estimation is shifted in the ordinate axis, and the same rule applies for the abscissa axis.

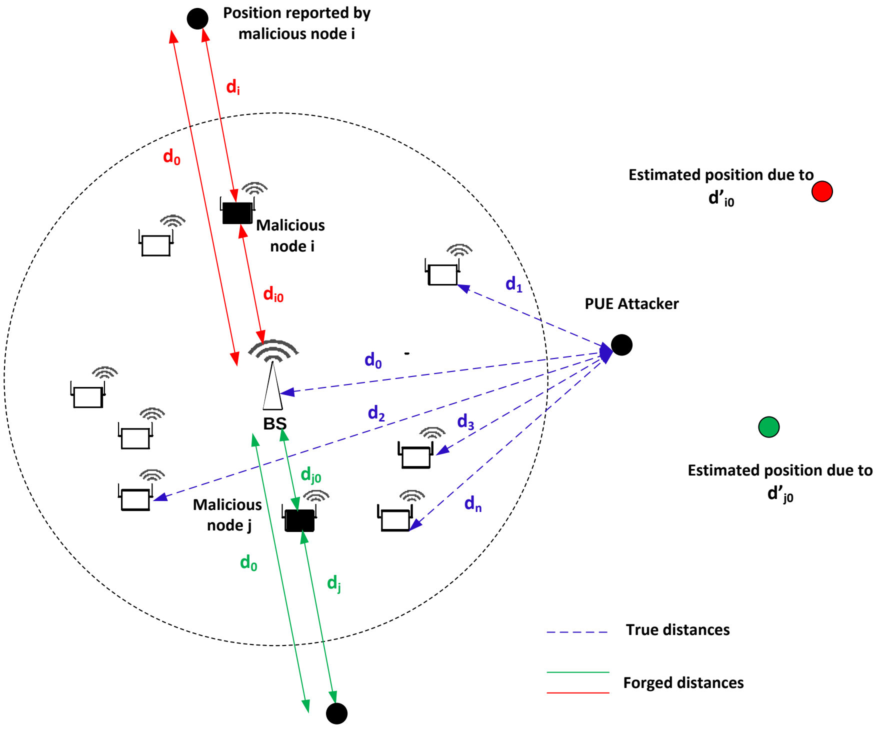


Figure 7.5: Effect of the position of the malicious node on the estimation

Figures 7.6 and 7.7 depict the error samples under the same scenario but with 2 and 3 liars respectively. It can be observed that, as the number of liars increase, the error samples tend to be more dispersed and with higher values. Obviously, the more number of incongruous measures introduced into the location method, the worst the accuracy of the estimation. Note also that the error is bounded to 120 kilometers in the ordinate axis. Because we have considered that primary users are not further than 150 kilometers from the BS, all position estimations exceeding this value have been discarded.

Besides the increase in the position estimation error, the presence of liars has another negative effect on the location method: lack of convergence. As some measurements are incongruous, the WLS algorithm is unable to find an optimal solution to the system of equations given by the TDoA measures (6.54). We also have considered as lack of convergence those simulations in which the estimated position is further than 150 kilometers from the BS. As mentioned above, we assume that true primary users will be always closer than 150 kilometers

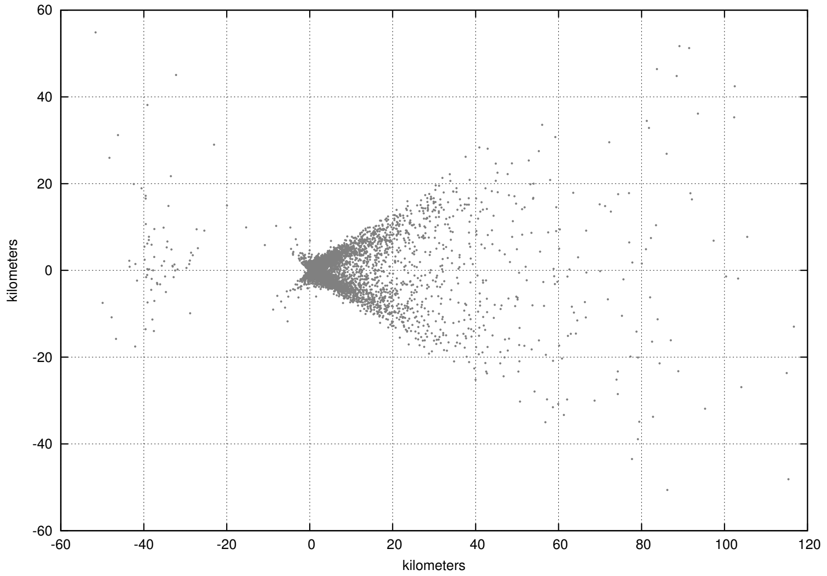


Figure 7.6: Prediction error with 2 liars (10000 experiments),(x,y)=(30000,0),n=30

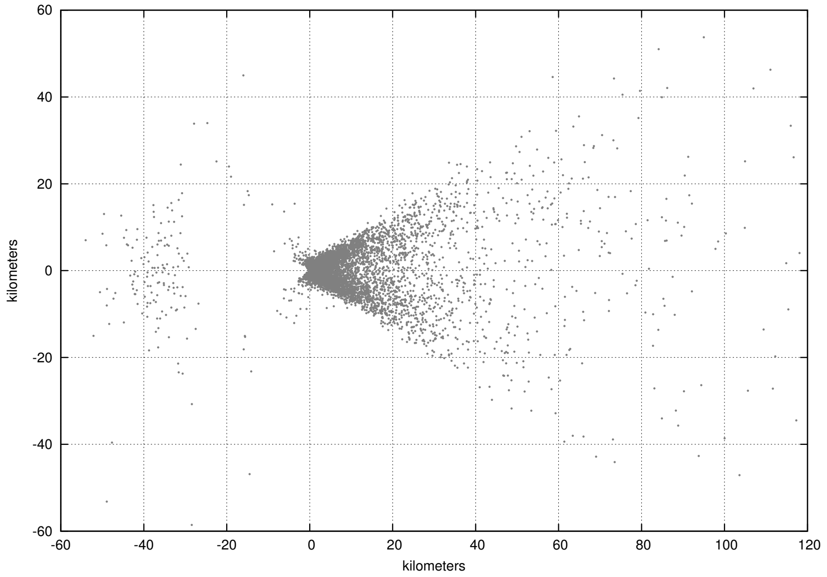


Figure 7.7: Prediction error with 3 liars (10000 experiments),(x,y)=(30000,0),n=30

to the BS and therefore any position estimation further than this distance is considered to be erroneous.

Table 7.2 shows the percentage of simulations, for each number of cooperating nodes n and liars, in which the WLS algorithm did not achieve convergence. As it can be observed, there are a considerable number of failures when the number of cooperating nodes is small, e.g., for $n = 10$ or $n = 20$ nodes. Obviously, this is because the ratio of liars to honest users is higher and thus forged measurements have a big influence on the location process. For values above $n = 50$ the probability of non-convergence is negligible and therefore we have omitted

N	No liars	1 liar	2 liars	3 liars
10	0	7.20	16.59	23.86
20	0	0.74	3.54	7.58
30	0	0.02	0.47	0.68
40	0	0	0.03	0.3
50	0	0	0	1

Table 7.2: Percentage of location failures

the corresponding results.

Impact of the position of the PUE attacker

As seen in Chapter 6, the accuracy of the location method strongly depends on the distance between the node to be located and the anchor nodes. Because of this, it is also necessary to analyze to what extent false measures can disrupt the location method depending on the position of the emitter.

Figure 7.8 depicts the error samples obtained when the PUE attacker is placed at position $(30000, 15000)$ and one out of $n = 30$ cooperating nodes is providing false feedback. With respect to the previous case in which the emitter laid at $(30000, 0)$, the 99% confidence interval error of the estimation has increased from 34.1 to 54.76 kilometers and the percentage of convergence failures has also risen from 0.02% to 0.44%. The degradation in the performance of the location method is due to the increase in the Euclidean distance between the emitter and the CRN. The distribution of the error samples are similar in both cases and are arrow-shaped, but it in the former case the arrow was aligned with the x axis, and now it is rotated several degrees. The reason for such effect is clearly the change in the position of the emitter, which has moved along the y axis to coordinate $y = 15000$m.

On the other hand, Figure 7.9 shows the error samples when the PUE attacker is positioned further from the CRN, at position (-75000,50000). The number of cooperating nodes is $n = 100$ and one of them is reporting forged measures to the BS. We do not present the results for $n = 30$ because for this position of the PUE attacker the location method does not achieve convergence.

Due to the huge increase in the distance between the CRN and the PUE attacker, false feedback has even a more harmful effect on the location method. The 99% confidence interval error of the estimation is of 72.35 kilometers, meaning that the location method is completely disrupted, and the percentage of convergence failures of 1.72%. As a consequence, the probability of a success PUE attack dramatically raises. Differing from the previous simulations, most of the samples are located on the negative axis due to the change in the position of the PUE attacker.

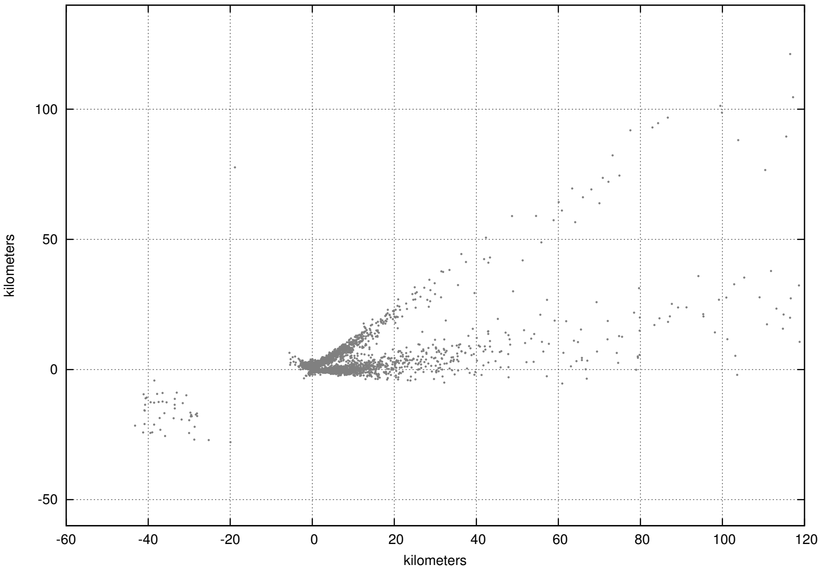


Figure 7.8: Prediction error with 1 liar (10000 experiments),(x,y)=(30000,15000),n=30

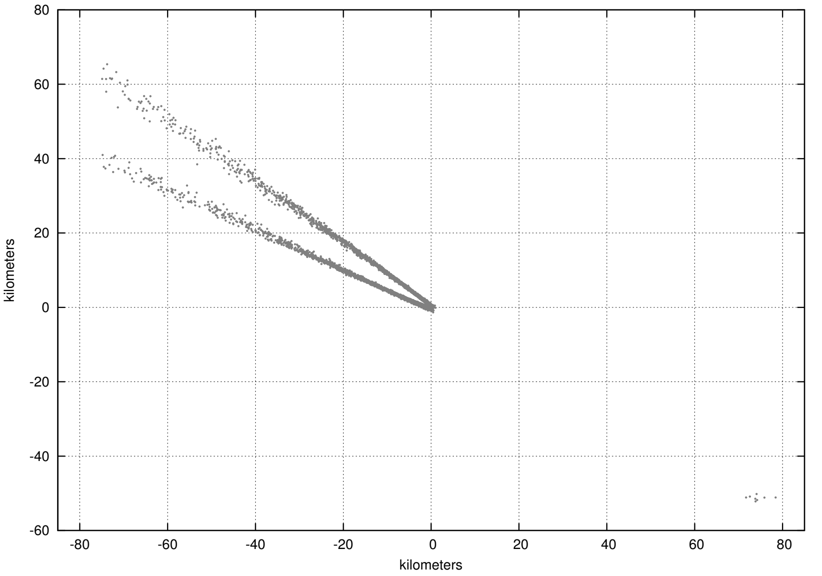


Figure 7.9: Prediction error with 1 liar (10000 experiments),(x,y)=(-75000,50000),n=100

7.3 Proposal of a cooperative location method resistant to liars

7.3.1 Motivation

As shown in the previous section, false reports provided by compromised nodes can severely undermine the location method, and thus can lead to false positives or false negatives regarding the detection of primary users. In this section, we propose a method for identifying false measurements and eliminate them from the location process, or at least reduce their impact on the location estimation. The method looks for deviations from the expected value in the measures provided by each CR. Based on the degree of deviation, each measure is assigned a weight which will determine how much this measure contributes to the location process.

In order to quantify to what extent a given measure deviates from the expected value, a reliable estimation of the position is needed. Let (x_v, y_v) be a reliable estimation of the emitter's position, and d_{i0} the difference distance derived from the TDoA measure for the pair BS and node i. Then, the deviation of every measure from the expected value, which we will refer to as residue r_i, can be computed as in 7.3:

$$r_i = d_{i,0} - \left(\sqrt{(x_v - x_i)^2 - (y_v - y_i)^2} - \sqrt{x_v^2 - y_v^2}\right) \tag{7.3}$$

However, to obtain a reliable estimation is challenging. *A priori*, all measures including those provided by malicious nodes are used as input to the location method, meaning that the estimation of the emitter's position will be distorted by forged measures. In order to overcome this problem, we propose a cluster-based method which makes use of the median as a parameter in order to identify false measures. This method is inspired on LMS fitting [Rousseeuw et al., 1987], a known technique typically used for outliers detection. With LMS, the position estimation (x_v, y_v) is computed as the solution which minimizes the median of the residue squares as in (7.4).

$$(x_v, y_v) = arg\,min\ [median_i\,(r_1, r_2, ..., r_i)] \tag{7.4}$$

The cost of computing the exact solution for (7.4) is prohibitive [Rousseeuw et al., 1987] and therefore an approximation can be obtained by means of clustering. The rationale behind it is as follows. If the set of CRs is divided into c clusters and there are at most $c-1$ liars in the network, at least one reliable estimation of the emitter's position can be obtained by applying the location method in each cluster separately. Considering the position estimated by a given cluster, it can be computed the difference between the measure provided by each node and the value it should have reported. Because the position estimated by clusters containing at least one liar will be less accurate, the differences computed for the nodes belonging to those clusters will be higher and therefore the median of these differences. As a consequence, based on the median of these differences, a reliable estimation can be obtained.

7.3.2 Procedure

In the following, we describe how to proceed in order to perform the location method robust to forged measures provided by malicious nodes.

1. Divide the set of n CRs into c several clusters of equal size $s = \lceil \frac{n}{c} \rceil$. Besides the BS, the minimum number of CRs in each cluster must be two, in order to be able to apply the location algorithm. However, a larger number of CRs is desired in order to

obtain an accurate estimation. Because the accuracy of the estimation improves with space diversity, that is, whenever measures are not correlated, clusters are generated by selecting a set of nodes randomly.

2. Apply the WLS method described in Section 6.2.3 separately in every cluster obtaining an estimation of the position of the emitter for each one. (x_{v1}, y_{v1})... (x_{vj}, y_{vj})...(x_{vc}, y_{vc}).

3. Compute the median of residue squares for each cluster j as

$$r^2_{cluster_j} = median(r_1^2...r_i^2...r_s^2)$$

with r_i the residue for node i of cluster j as in (7.5).

$$r_i = d_{i,0} - \left(\sqrt{(x_{vc} - x_i)^2 - (y_{vc} - y_i)^2} - \sqrt{x_{vc}^2 - y_{vc}^2}\right) \tag{7.5}$$

4. Select as tentative estimation (x_v, y_v) the one given by the cluster with the lowest median of residues squares.

5. Compute the residue squares for all the n nodes considering the tentative estimation (x_v, y_v).

6. Perform a new position estimation by applying the location method assigning a different weight to each node's measurement according to its residue square.

If we assume that at least in one of the clusters there are no false measurements, the WLS estimation provided by the cluster will be more reliable and will exhibit a lower median of residues squares. Thus, when computing the residues in step 4 with this estimation, false measurements will be clearly identified due to its higher residue with respect to the rest. Since the accuracy of the location method improves as the number of (reliable) measurements increases, a final WLS estimation is performed in the last step excluding abnormal measurements or reducing their effect on the estimation by assigning them a lower weight.

The number of clusters plays an important role on the method since it determines the number of liars l to which the location process will be robust. However, in a l-resistant method, the WLS algorithm must be applied $l + 1$ times: one for each cluster and another for the final estimation where measures are weighted according to its residues. This fact can introduce significant overhead with regard to computation time if the system is not properly designed, i.e., if the number of real liars is much lower than the factor of resistance of the location method.

Assigning weights to nodes' measures

In order to apply the WLS method, it is needed to define a diagonal matrix W where its non zero elements w_{ii} are the weights assigned to each measure. As previously outlined, we compute each weight according to the residue value of the measure provided by the node. In order to quantify how probable is a given residue, we make use of the pdf of the measure error for each node. Assuming that measurement errors E are normally distributed with zero mean and variance σ_i computed as in (6.62), the probability that the error takes a given value $E = \varepsilon$ is given by its pdf evaluated at that value $\texttt{pdf}(E = \varepsilon)$. This probability is assumed to be an indicator of the reliability of a given measure and therefore, the weight assigned to it is computed as in (7.6).

$$w_{ii} = \frac{\texttt{pdf}(E = \varepsilon)}{\texttt{max}\left(\texttt{pdf}(E)\right)} \tag{7.6}$$

The reliability indicators are normalized so that $w_{ii} \in [0, 1]$. In this way, a measure will be assigned a weight $w_{ii} = 1$ whenever its residue is $r_i = 0$, that is, when it completely matches the position estimation (x_{vc}, y_{vc}) of the cluster to which it belongs.

7.3.3 Evaluation

Aiming to evaluate the goodness of the proposed method, we have run a set of 10000 simulations considering the same scenario as in Section 7.2.2, i.e., a 802.22 network with n cooperating nodes and a PUE attacker placed at $(30000, 0)$. We also have considered the existence of one or two liars out of n nodes.

In the following, we present the simulation results obtained when applying the clustering method. These results are compared to those obtained by means of the conventional location method under the presence of the same number of liars. In addition, we also provide the simulation results showing the effect of the clustering method on the estimation accuracy when there are no malicious nodes providing false feedback. Because there is no a *a priori* knowledge of the number of liars, the network must select the number of clusters before performing the location process. As a consequence, it is also important to analyze the behavior of the method when there are no liars.

Case 1: 1 liar

Table 7.3 shows the 99% confidence interval error of the estimation for the 10000 simulations under the presence of one liar and when the clustering method is applied, with 2 clusters. For ease of analysis, the table also provides the results for the case in which the conventional location method is applied, already presented in the previous section.

As it can be observed, the proposed mechanism can effectively identify deviated measurements and the error performed in the estimation is reduced more than a 98% with respect to the conventional method. In particular, for a number of cooperating nodes above $n = 80$

n	No liars		1 liar	
	no clusters	2 clusters	no clusters	2 clusters
10	358.9	744.29	112000	1593.8
20	75.04	246.78	66779	307.98
30	46.64	132.94	34105	139.23
40	35.28	80.28	20948	87.56
50	30.06	53.96	12642	64.68
60	26.12	42.8	9626	46.58
70	23.4	37.51	8087	37.54
80	21.24	32.2	6562	32.17
90	19.85	28.42	5078	29.92
100	18.82	27.05	4442	25.9

Table 7.3: 99% confidence interval error (meters) in the presence of 1 liar

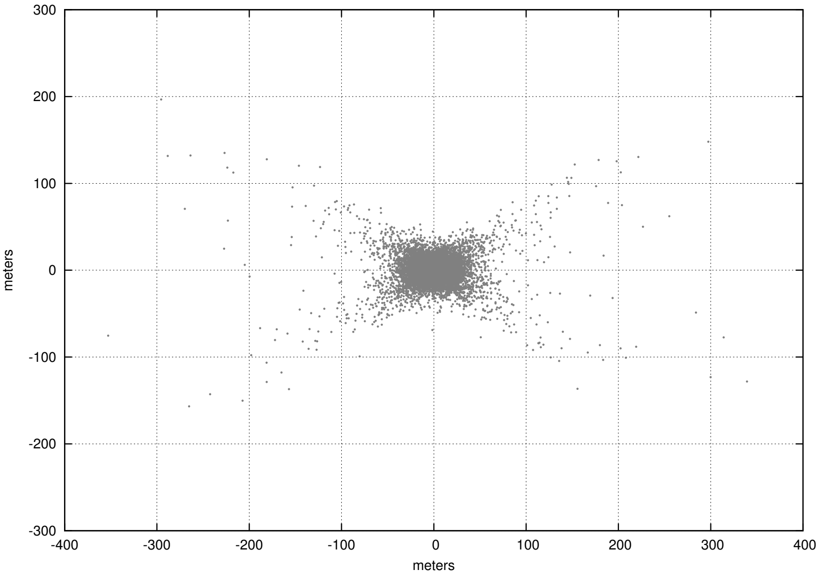


Figure 7.10: Prediction error with 1 liar and 2 clusters (10000 experiments), (x,y)=(30000,0), n=30

nodes, the location process is performed almost with the same accuracy as in the case when there are no liars. When the number of cooperating nodes is small, the error is significantly higher than in the case with no liars (between 50% and 300% higher), but still on the order of a few tens of meters. Therefore, its accuracy is still enough to distinguish the PUE attack from a real primary user.

Figure 7.10 shows the error samples obtained for $n = 30$ under the presence of one liar and when 2 clusters are applied. It can be seen that samples are more dispersed than in Figure 7.3, i.e., the case in which there are no liars and no clustering method is applied. This is due to weights' mechanism applied by the clustering method.

As expected, the poorest results are given by the case $n = 10$. On the one hand, the conventional method is already less precise for such a small number of cooperating nodes. Moreover, by applying the clustering method, a position estimation is obtained for every cluster with only 5 measures. This implies that, even a cluster containing no liars will provide estimations with low accuracy. On the other hand, when our method is applied but no CRs are providing false feedback, the accuracy of the estimation gets slightly worse than with the normal procedure, again due to the weight assigned to each measure. Because weights are assigned to each measure according to their residue and taking into account the best cluster estimation, honest measures may be given low weights if the accuracy of the cluster estimation is not good enough. With a high probability, this will happen whenever the number of cooperating nodes n is small, since as mentioned before, this leads to small clusters which will not be able to obtain accurate estimations.

Figure 7.11 shows the error samples obtained for $n = 30$ when there are no liars and 2 clusters are applied. The distribution of the samples is very similar to that shown in Figure 7.10. As mentioned above, the accuracy slightly gets worse when the clustering method is applied and the number of cooperating nodes is small.

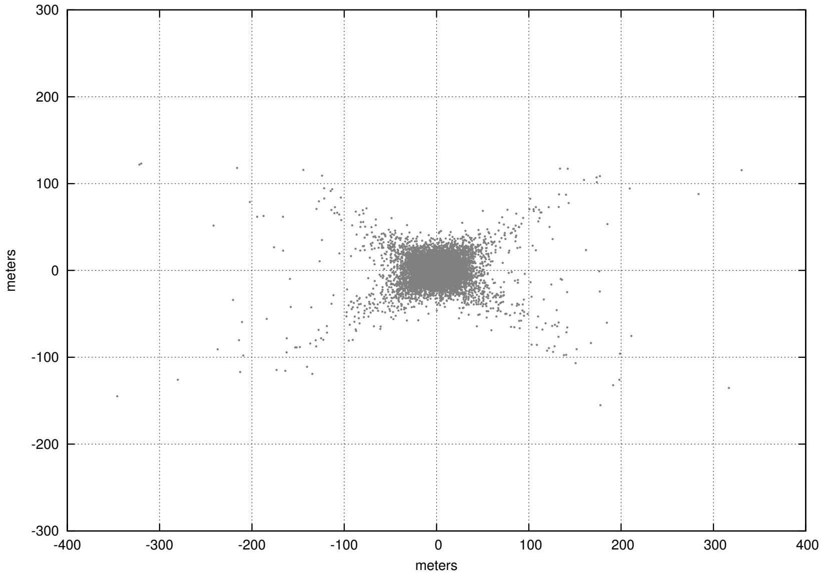


Figure 7.11: Prediction error with no liars and 2 clusters (10000 experiments), (x,y)=(30000,0), n=30

A second improvement of the proposed method is that achieves a higher rate of convergences Table 7.4 shows the percentage of failures both for the conventional method and for the clustering method, when one or no liars are present. The results for a number of cooperating nodes above $n = 50$ have been omitted since the location method always achieves convergence. As it can be observed, for a number of cooperating nodes $n = 10$ with one liar, the clustering method exhibits some failures while the conventional method doesn't. This is again due to the fact that the generated clusters contain only 5 nodes, and either the WLS method is not

N	No liars		1 liar	
	no clusters	2 clusters	no clusters	2 clusters
10	0	0.05	7.2	2.26
20	0	0	0.74	0.44
30	0	0	0.47	0.16
40	0	0	0.03	0.01
50	0	0	0	0.01

Table 7.4: Percentage of location failures in the presence of 1 liar

able to find a solution or the estimation position exceeds the distance limit (150 kilometers).

Case 2: 2 liars

When there are 2 out of n cooperating nodes providing false feedback, at least 3 clusters are required in order to guarantee that abnormal measures will be detected. Table 7.5 shows the 99% confidence interval error of the estimation for the 10000 simulations under the presence of 2 liars and when the clustering method is applied. Because the network may be configured to be robust only to one liar, it is also needed to analyze the behavior of the method when a 2-clustering method is applied. The corresponding results are also shown in Table 7.5. As it can be observed, when 3 clusters are applied and there are 2 liars, the estimation error is reduced more than a 98% for all numbers of cooperating nodes except for $n = 10$. In this case, the method does not provide robustness and fails in obtaining a reliable estimation. This is because clusters are provided only with 3 measurements and the WLS algorithm is not capable of finding an optimal estimation.

Also note that, if only 2 clusters are applied, the error is considerably higher than when the number of cooperating nodes is below $n = 40$. It is a quite reasonable result, since in such case,

N	No liars		2 liars		
	no clusters	3 clusters	no clusters	2 clusters	3 clusters
10	358.9	907	117960	27255	25242
20	75.04	234.39	98175	23018	433.45
30	46.64	131.37	53794	11875	171.71
40	35.28	86.18	34933	872.04	98.19
50	30.06	54.19	22165	169.23	63.1
60	26.12	45.67	15359	63.75	46.76
70	23.4	36.54	11094	43.59	39
80	21.24	33.95	9038.6	37.07	34.17
90	19.85	29.26	7671	29.54	29.51
100	18.82	27.04	6066	28.4	27.14

Table 7.5: 99% confidence interval error (meters) in the presence of 2 liars

an accurate estimation can be obtained only when the two liars belong to the same cluster, and the remaining cluster performs the location method without false data. However, when every liar belongs to a different cluster, the situation is equivalent to having two clusters with one liar of at most $n = 20$ CRs. Looking back to Table 7.3, we can see that the 99% confidence interval error of the estimation for such case is around 66 kilometers. This implies that even the best estimation out of the two clusters will be completely erroneous, leading to high residues for all measures. As the weights assigned to each measure for the final location process are computed according to the pdf of the error measurements, which we assume to be normally distributed, both true and forged measures will obtain a very low weight. On the other hand, for a high number of cooperating nodes the error is gradually reduced. For example, the 99% confidence interval error for $n = 50$ is of 12.64 kilometers. Since the estimation given by the best cluster is closer to the real position of the emitter, the weighting scheme will assign a much lower reliability indicator to forged measurements, and the final estimation will be more accurate.

Table 7.6 shows the percentage of failures when 2 liars are present and 2, 3 or no clusters are applied. The number of failures considerably increases with respect to the case of 1 liar, in particular when the number of cooperating nodes is small. Again, the rate of convergences is higher when the conventional method is applied than when the robustness parameter is not well selected, i.e., when the number of clusters is 2. However, it can be somehow considered as an improvement, since it is preferable to obtain a convergence failure than a wrong estimation of the position, which could lead to a false positive.

N	No liars		2 liars		
	no clusters	3 clusters	no clusters	2 clusters	3 clusters
10	0	0	16.66	34.67	4.13
20	0	0	3.54	15.40	0.49
30	0	0	0.47	5.33	0.18
40	0	0	0.03	0.18	0.04
50	0	0	0	0.64	0.01
60	0	0	0	0.17	0.01
70	0	0	0	0.11	0.01
80	0	0	0	0.01	0
90	0	0	0	0.01	0
100	0	0	0	0.01	0

Table 7.6: Percentage of location failures in the presence of 2 liars

The particular case for small values of n

The number of clusters can be critical when the amount of cooperating nodes is small. First, it should be guaranteed that there is at least one cluster without liars. Moreover, in order to achieve a minimum level of accuracy in the estimation performed by each cluster, the number of nodes in a cluster should be as high as possible. However, this can represent a problem when the total amount of cooperating nodes is small. For example, for $n = 10$, 3 clusters are required if robustness to two false measurements is desired. This leads to only 3 available measurements for each cluster. Because the less the measurements introduced into the location process, the lower the accuracy of the estimation, this situation could lead to inaccurate estimations in all the clusters and therefore to a wrong final estimation of the emitter's position.

In order to partially overcome such limitation and improve cluster's estimations, the following approach can be adopted. Consider n cooperating nodes and a location method robust to l liars, i.e., $c = l + 1$ clusters must be formed. In order to guarantee that at least one cluster is free of liars, there must be at most $n - l$ elements in each cluster. For example, if $n = 10$ and $l = 2$ liars are tolerated, $c = \binom{n}{n-k} = \binom{10}{(10-8)} = 45$ different clusters can be generated, each one containing 8 elements. In this way, the above mentioned requirements are satisfied: the number of measures for performing the location method is higher and there will be at least one cluster free of liars.

In the following, we show the improvement introduced by such clustering method. We provide the results for a set of simulations considering again a PUE attacker placed at $(30000, 0)$ and a number of cooperating nodes $n = 10$. Out of the 10 cooperating nodes, 1 or 2 are providing false feedback. Table 7.7 depicts the 99% confidence interval error of the position estimation, and the percentage of failures. For the scenario with one liar, we provide the simulation results obtained by generating $c = 2$ clusters with 5 nodes in each cluster, i.e., the standard method, and also by generating $c = \binom{n}{n-k} = \binom{10}{(10-1)} = 10$ clusters, each one with 9 nodes. Analogously, for the case with two liars the results correspond to the generation of $c = 3$ clusters and $c = \binom{n}{n-k} = \binom{10}{(10-8)} = 45$ clusters with 8 nodes in each one.

As it can be seen, the method achieves a reduction of the estimation error of a 56.45% for the case with 1 liar. When there are 2 liars, the error is reduced a 96.68%. This improvement comes, however, at the expenses of a high computation overhead since the WLS method must be applied to every cluster independently. For this reason, it should be applied only in those cases in which the network has evidence of the existence of any liar node.

Note also that this technique could be somehow applied in scenarios with higher values of n in order to improve location accuracy. In some cases the location method may not converge due to the specific way in which nodes have been grouped. If this occurs, the location method can be applied again by generating a new set of clusters. Despite by means of this technique the probability of convergence can be increased, it can lead to long run

1 liar				2 liars			
Error		Failures		Error		Failures	
c=2	c=10	c=2	c=10	c=3	c=45	c=3	c=45
1593	604.60	2.26	0.7	25242	836.57	4.13	0.54

Table 7.7: 99% confidence interval error (meters) and percentage of failures for small n

times. By way of illustration, consider a network with $n = 90$ cooperating nodes which are divided into 3 clusters with $k = 30$ node in each one. These nodes can be grouped into $\binom{n}{n-k} = \binom{90}{90-3} = \frac{90!}{30!(90-30)!} \approx 6.73 \cdot 10^{23}$ different ways. As a consequence, it should be set an upper bound to limit the number of attempts.

7.4 Conclusions

In CRNs, the presence of malicious users providing false data represent a serious threat to those mechanisms relying on the cooperation of the members of the network. One of these mechanisms is the cooperative detection process described in Chapter 6, which based on a set of TDoA measures derived from data provided by the CRs and the BS of the network, estimates the position of a given emitter by applying a WLS method. The accuracy of the method allows distinguishing between a legitimate primary user and a PUE attacker, whenever the cooperating nodes are honest and report true measures.

In this chapter, we have analyzed the effect of misbehaving nodes which deliberately forge measures in order to lead the CRN to a wrong estimation of the emitter's position, further from the network than the real one. This attack is targeted to increase the probability of a successful PUE attack. The simulation results show that, with only one single node reporting false feedback, the error performed in the location estimation dramatically increases, particularly when the total number of cooperating nodes is small.

In order to mitigate the effects of such attack, we have proposed a method resistant to l liars based on minimizing the LMS error. The method requires to divide the set of CRs into $c = l + 1$ different clusters and apply a WLS based location process separately in each cluster. Based on the position estimation given by each cluster, the error residue of each node and the median of the residues are computed for each cluster. In this way, those clusters containing any liar are clearly identified because they exhibit a higher value of the median. Then, the one with the lowest value is considered to be the cluster providing the best estimation. Finally, this cluster estimation is used to detect false measures by computing the error residues for all nodes. A reliability indicator for each measure is computed based on the probability of a given node providing a measure with such error. The final position estimation is obtained by applying again a WLS method, where each measure is weighted according to its reliability indicator. In this way, the effect of false measures on the estimation is diminished.

The results obtained show that this method outperforms the conventional location scheme under the presence of liars. In fact, estimation errors are dramatically reduced and are comparable to those obtained in the absence of false feedback. However, the complexity of the location process increases with the level of robustness since the WLS method must be applied $l + 1$ times in order to be robust to l liars. We have also analyzed the performance of the method when the level of robustness is higher than the the number of liars. In such case, the accuracy of the estimation slightly diminishes in comparison to the basic location method.

Despite it, it is still worth to apply the method in order to prevent potential attacks.

The application of this mechanism poses a new challenge, since a CRN may not know in advance the number of liars present in the network. This implies that in some situations the network may perform more computations than those needed and lose a bit of accuracy in the position estimation. Because of this, an initial guess of the number of liars l is needed in order to design the mechanism. To address this problem, we suggest to use a reputation system. As compromised nodes are likely to report false data repeatedly, a trust mechanism could be integrated into the location process, so as to keep track of node's behavior over time.

Trust and reputation models have been extensively studied, especially in the context of ad hoc networks [Mejia et al., 2009]. Recently, the idea of applying them to CRNs to enhance collaborative spectrum sensing has attracted research interest [Qin et al., 2009]. In a similar way, nodes' reputation could be applied to the location process of an emitter, both to determine the number of clusters to be used and to assign weights when applying the WLS method. Moreover, they could be also used to refine the method and improve its accuracy by generating clusters according to the reputation of each node. Those nodes with higher reputation could be grouped into the same cluster, not only allowing to obtain a very accurate cluster estimation but also more precise reliable indicators, and therefore a better final estimation of the emitter's position.

Part IV

Conclusions and Future Work

Chapter 8

Conclusions and Future Work

The present chapter concludes this dissertation and is therefore intended to summarize the most salient results of this work but also to foresee future steps.

8.1 Conclusions

CRNs are envisioned as a possible solution to the current inefficiency in spectrum usage. These networks are composed of intelligent devices with cognitive and reconfigurability capabilities. Based on medium sensing and past experiences, these devices or CRs are able to detect spectrum holes and determine the optimal transmission parameters such as frequency, coding, modulation, etc. According to such information, CRs can tune themselves in order to maximize network performance. Due to these properties, CRNs have been proposed to access in an opportunistic way and on a non-interfering basis to those portions of licensed spectrum left unused by their legal owners, i.e., primary users such as TV broadcast systems. In order to realize the CR paradigm, however, it is needed to properly design effective mechanisms to accomplish the different cognitive functionalities.

Throughout this thesis we have explored those threats which could severely undermine the performance of CRNs and potential countermeasures to effectively deal with them. The first contribution of this work has been an analysis of the State of the Art with regard to security in CRNs. CRNs can be deployed in many ways leading to different architectures, each one making the network more vulnerable to particular threats. For this reason, the different attacks reported in the literature have been studied from a critical point of view, and a qualitative analysis of their impact on the different CRN architectures has been provided. Analogously, we have taken a look to the existing countermeasures and have analyzed their effectiveness.

This preliminary work led to the design of a roadmap, which was intended to provide future researchers with a draft of the mechanisms needed in order to counteract the main threats to CRNs. The performance of CRNs is especially sensitive to erroneous sensing data, either due to liar nodes contributing to the sensing process with forged reports or due to malicious users altering the medium. For this reason, this roadmap specially focuses on the PUE attack and FF attacks. In the former, an attacker intentionally transmits a signal pretending to appear as a primary user in order to force a frequency handoff and prevent a CRN from using a vacant band. FF attacks are addressed to disrupt any cooperative mechanism used in a CRN by providing false information to the entity in charge of taking a given decision, e.g., the decision about the existence of a primary user in a given channel based on a set of measurements provided by a set of CRs.

This roadmap outlines the modules that should be included in any security system for CRNs and describe the main features of each one. Furthermore, it has been pointed out the need for a flexible security system in order to accommodate other modules to counteract new potential attacks. This draft constitutes the second contribution of this thesis, and the basis for the remaining work here presented.

In CRNs, there is a strong relationship among the different layers of the TCP/IP model. For example, the environment information gathered at the PHY layer has a direct impact on the spectrum management and spectrum mobility functionalities performed at the MAC layer. In their turn, the decisions taken at the MAC layer influence to a large extent the performance of upper layers, such as routing functions at the IP level or reliable delivery of data implemented at the transport layer. Because of this, cross-layer attacks can represent a big threat to CRNs.

The third contribution of this thesis has been the proposal and the modeling of a cross-layer attack to CRNs coined the Lion attack, which is performed at the PHY layer and is targeted to disrupt the performance of TCP connections. In this attack, a malicious user forces a frequency handoff in the CRN, for example by means of a PUE attack. As a consequence, all communications in the network are interrupted until a new channel is available leading to TCP data loss and expiration of the retransmission timers. These facts are considered as a signal of congestion by the TCP protocol which reacts to them by reducing its transmission rate.

The impact of the attack is evaluated by measuring the amount of time that a TCP sender remains inactive, i.e., without transmitting any data, as a consequence of such attack. The value of the inactivity time has been obtained by means of simulation but also through an analytical model, which has been derived for the attack. The provided results show that, due to the congestion control mechanisms used by TCP, the attack can lead to long periods of inactivity even when a channel is available for transmission. In addition, we have presented a smart version of the attack in which the attacker forces frequency handoffs coinciding with the retransmission attempts of the TCP sender. It has been shown that in such case the Lion attack leads to a DoS.

Several countermeasures have been outlined in order to deal with this attack. On the one hand, we have proposed a modification to the TCP normal operation in order to mitigate the effect of the attack. Such proposal is based on freezing the TCP congestion control mechanisms during a handoff, and thus it requires cross-layer exchange of information between the PHY and the transport layers. The effectiveness of this countermeasure has been evaluated via simulation, proving that it allows a TCP sender to take profit of the bandwidth while a given channel is available. Such countermeasure is addressed to minimize the impact of the attack on TCP throughput but cannot either detect or avoid the attack. For this reason, a location method is suggested with the purpose of detecting the attack based on the position estimation of the source transmission.

The fourth contribution of this thesis has precisely been the development and evaluation of a cooperative location scheme, which is intended to protect 802.22-based CRNs against PUE attacks based on the emulation of TV broadcast systems. Most proposals address the problem of detecting PUE attacks by measuring the energy level of the received signal at several CRs. However, such approaches can easily be circumvented by an attacker by adapting its power transmission. Our proposed method also benefits from the cooperation of CRs but differs in two fundamental aspects: first, it relies on TDoA measurements provided by the set of CRs and second, the target is to estimate the position of the emitter. When primary users are TV towers, it can be assumed that their position is known, and therefore the CRN can detect whether a given transmission is an attack or not by comparing the estimated position with those of primary emitters. The use of TDoA measurements for the location process allows to achieve higher accuracy in the estimation but on the other hand, it poses some challenges that we have effectively addressed in our proposal, such as the need for synchronization among CRs. On the other hand, in order to minimize the error performed in the estimation, a WLS method is applied.

The accuracy of the location method has been evaluated by means of simulation under different scenarios. The error performed in the estimation strongly depends on the distance between the emitter and the CRN, and also on the SNR at the receivers performing the measurements. Generally speaking, this proposal can effectively detect PUE attacks and avoid false positives regarding spectrum availability. The best chance for an attacker is to perform the attack from a position near a primary user and control its transmission power to ensure a low SNR at CRs, i.e. -10dB or less. In such case the error performed in the estimation can go up to a few kilometers, increasing the risk of false positives. Despite it, in 802.22 networks, primary users are expected to be around 150 km far away from the CRN, meaning that an attacker should have a transmission power similar to a TV tower to successfully perform such attack. Besides, under such conditions the attacker would also interfere TV transmissions, therefore representing a higher threat to the primary network than to the CRN. As stated by the FCC [Security and Council, 2004], TV stations must be protected by means of security personnel, video surveillance and other methods. As a consequence, it would be rather improbable to successfully perform a PUE attack in the vicinity of a TV tower.

The location method relies on the cooperation of CRs reporting TDoA measurements and, as a consequence, it is vulnerable to malicious CRs providing false data, i.e., liar nodes. The last contribution of this thesis has been the proposal of a mechanism in order to identify forged measurements and make the location process robust to them. This mechanism requires the set of CRs to be grouped into a given number of clusters and to apply an independent location process for each cluster. The median of the residues i.e., the difference among the value reported by a given node and the value it should have reported, is used to determine the cluster providing the best estimation. According to such estimation, the residues for all nodes are computed and those presenting an abnormal high residue are regarded as liars. A final location process is performed assigning a different weight to each measure depending on its residue, so that the contribution of forged measurements to the final estimation is minimized. A key parameter of this mechanism is the number of clusters c, which defines the number of liars $c - 1$ to which the location process is robust.

The evaluation of the method by means of simulation shows that, when the number of clusters is higher than the number of liars, it can effectively deal with forged measures and provide position estimations with the same accuracy as if there were no liars. However, this comes at the expenses of additional computations. Therefore, the method should be properly designed, regarding the value of c, in order to avoid an unnecessary waste of resources when there are no liars or less than expected.

8.2 Further work

In this section, we discuss some aspects of this thesis which could be further improved but also propose new lines of research which still remain open after this work.

With regard to PUE attacks, we have focused on those based on TV broadcast systems. However, in the IEEE 802.22 standard, wireless microphones are also considered as primary users. Dealing with PUE attacks based on wireless microphones is still challenging, since they are low-power transmit devices and, differing from TV towers, their location can vary over time. The 802.22 standard proposes the use of beacon signals in order to signal the presence of wireless microphones which could also carry authentication information, e.g., digital signatures. Although such mechanism could indeed allow to distinguish between legitimate and fake transmissions, it is not expected that in a near future all legacy microphones will meet these requirements [Chen et al., 2011]. Thus, alternative mechanisms should be provided in the meanwhile.

In view of the fact that the position of wireless microphones is not known in advance, it seems that at first glance, the location method we have proposed in Chapter 6 would also fail in detecting such attacks. A possible solution could be the use of a scheme based on misuse detection, as we already suggested in Section 4.2.3. Misuse detection mechanisms are based on the definition of signatures which model different attacks. The rationale behind it is that well-known attacks can be detected by keeping track of the events taking place in the CRN, and looking for that specific signatures during the operation of the network. For a malicious user performing wireless microphone-based PUE attacks each time the CRN switches to a new channel, the signature would be given by its location (obtained by means of the location method proposed in this thesis) and the fact that such potential primary user is always transmitting at the same frequency as the CRN.

Concerning the proposed location method, we have assume that PUE attackers are static. Therefore, a natural next step forward would be to adapt our proposal to support the mobility of PUE attackers. Then, a location technique employing motion characteristics, such as the EKF, would be probably more effective to locate the attacker than the Taylor-Series estimation algorithm used in our proposal. On the other hand, RFF techniques could help to associate attacks performed at different locations with a single malicious device.

Moreover, in our simulations we have always considered a stationary network where the anchor nodes are static and with known positions. Because the mobility of anchor nodes would strongly affect the accuracy of the location method, a performance analysis of such scenario and possible improvements would be also another future line of research.

The provided simulation results have been obtained assuming a transmission channel where effects such as multipath have not been taken into account. As a consequence, the performance of the proposed location method should be also evaluated under the errors produced by multipath. Although such errors are difficult to model because are dependent on the terrain where the network has been deployed, there are some models [Greenstein et al., 1997, Aso et al., 2001] which provide an analytical expression for the effect of multipath on time measurements, as a function of the distance between the emitter and the receiver.

In reference to the liars detection method, the next step forward would be the design of a trust mechanism in order to keep track of nodes' behavior over time with respect to the reliability of their measures. The trust values provided by such mechanism could then be used with two different purposes: first, to perform the location process assigning a different weight to each node according to its trust and second, to select the number of clusters c taking into account the number of potential liars, i.e., nodes exhibiting lower values of trust.

On the other hand, the adaptation of the proposed methods to fully distributed networks is another interesting line of research that we have considered to follow in the short run. The lack of a central entity controlling the feedback provided by CRs poses new challenges regarding several issues such as detection of selfish and liar nodes.

Finally, in this thesis we have mainly focused on those threats targeted to the sensing mechanisms of CRNs since disruption of such mechanisms has a negative impact on the remaining functionalities. However, there is still an entire set of vulnerabilities to be addressed, and new ones that will be discovered as CR technology evolves. This vulnerabilities may be related not only to spectrum sensing but also to other mechanisms such as spectrum sharing or spectrum management. As an example, in a CRN the available spectrum must be fairly distributed among its participants according to their requests. A malicious node could send fake requests in order to mislead the sharing mechanism, leading to loss of opportunities for well-behaved nodes, or could make an unauthorized use of the spectrum with selfish purposes. Because of this, further work is also needed in this direction.

Acronyms

ACK Acknowledgment. 62–64, 67, 77

AES Advanced Encryption Standard. 16

AI Artificial Intelligence. 26

AoA Angle of Arrival. 53, 88, 91, 92

ATSC Advanced Terrestrial System Committee. 9, 120

BER Bit Error Rate. 76

BS Base Station. 7, 8, 11, 13–16, 18, 28, 30, 31, 40, 51, 53, 55, 76, 87, 88, 92, 101–106, 108, 109, 111, 116, 118–122, 125–130, 132–134, 136, 144

BWS Bi-Weight Scale. 43

CBP Coexistence Beacon Protocol. 16

CCTT Channel Closing Transmission Time. 15, 16

cdf cumulative distribution function. 69, 70

CDP Cumulative Distribution Probability. 108

CDT Channel Detection Time. 15

CMT Channel Move Time. 15, 16

CP Cyclic Prefix. 14, 121

CPE Consumer Premise Equipment. 13–16

CR Cognitive Radio. iii, 4–6, 8, 10, 11, 13, 16–18, 23, 25–28, 30, 35, 36, 38–40, 42, 44, 45, 51, 53–55, 58, 59, 83, 92, 101–103, 106–109, 112, 114, 119, 122, 125–127, 129, 130, 136, 142, 144, 149–153

CRLB Cramer-Rao Lower Bound. 105

CRN Cognitive Radio Network. iii, iv, vi, 4–8, 11, 13, 16–19, 23–26, 28, 30, 31, 35–45, 49–56, 58, 61, 66–70, 75–77, 79–81, 83, 84, 87, 89, 90, 92, 101–103, 108, 111, 112, 114, 116, 118, 119, 122, 123, 125–127, 129, 131, 134, 144, 145, 149–153

cwnd congestion window. 62–64

dB decibel. 89, 109, 110, 112, 114, 116, 118

dBm dB milliwatt. 89

DFS Dynamic Frequency Selection. 15

DL Downlink. 14, 120

DoS Denial of Service. iv, 23–25, 28, 31, 32, 39, 52, 56, 61, 66, 75, 83–85, 150

DSA Dynamic Spectrum Access. 3, 4

DTV Digital Television. 120, 121

ECC Elliptic Curve Cryptography. 16

EGC Equal-Gain Combining. 12

EIRP Equivalent Isotropically Radiated Power. 13, 76

EKF Extended Kalman Filter. 99, 153

EMS Electromagnetic Signature. 38

FCC Federal Communications Commission. 3–5, 13, 15, 36, 38, 151

FF False Feedback. 30, 50, 51, 55, 56, 127, 128, 130, 150

FIFO First In First Out. 33

FTP File Transfer Protocol. 76

GCC Generalized Cross-Correlation. 90

GCM Galois Counter Mode. 16

GKM Group Key Management. 44

GPS Global Positioning System. 88

HTTP HyperText Transfer Protocol. 32

i.i.d. independent and identical distributed. 12

IDS Intrusion Detection System. 44, 49

IEEE Institute of Electrical and Electronics Engineers. 8, 13, 16, 25, 31, 40, 42, 56, 87, 89, 92, 102, 106, 108, 111, 152

IETF Internet Engineering Task Force. 83

IP Internet Protocol. 32, 150

ISI Inter-Symbol Interference. 13, 14

JF JellyFish. 33

LE Learning Engine. 39, 50, 52, 57, 58

LKF Linear Kalman Filter. 97, 99, 100

LMS Least Median Squares. iv, 128, 136, 144

LoS Line of Sight. 90

LS Least Squares. 97, 98, 104, 122, 128, 144, 151

LV Location Verifier. 37

MAC Medium Access. 13, 15, 16, 150

MAD Median Absolute Deviation. 43

MHz megahertz. 9, 13–15

MPDU MAC PDU. 16

MRC Maximal-Ratio Combining. 13

MSS Maximum Segment Size. 63

NPRM Notice of Proposed Rule Making. 4, 13

OFDMA Orthogonal Frequency-Division Multiple Access. 13, 14, 120, 121

OS Operative System. 65

P2P Peer-to-peer. 7

pdf probability density function. 69, 70, 138, 142

PDU Protocol Data Unit. 15

PHY physical. 13, 150, 151

PKC Public Key Cryptography. 43, 49

PKM Privacy Key Management. 16

PUE Primary User Emulation. iv, 18, 19, 25, 26, 28, 35–39, 50, 52, 53, 56–59, 66, 67, 75, 84, 85, 87, 90, 101, 102, 108, 112, 114, 116–119, 122, 123, 125, 126, 129, 130, 134, 138, 139, 143, 144, 150–153

QAM Quadrature Amplitude Modulation. 13, 121

QoS Quality of Service. 4, 14, 31

QPSK Quadrature Phase-Shift Keying. 13, 122

RC4 Rivest Cipher 4. 29

RF Radio Frequency. 4, 38, 53, 88, 90

RFF Radio Frequency Fingerprinting. 37, 38, 56, 57, 153

RMSE Root-Mean-Square Error. 108

RSA Rivest-Shamir-Adleman. 16

RSS Received Signal Strength. 36, 37, 53, 88, 89, 91, 93, 94, 96, 101, 119, 123

RST Reset. 32

RTO Retransmission TimeOut. 33, 63–65, 67, 68, 70, 71, 77

RTS RTT sample. 64

RTT Round-Trip Time. 63–65, 67, 68, 70, 76

RTTVAR Round-Trip Time Variation. 64

rwnd receiver window. 62

SCC Simple Cross-correlator. 89

SDR Software Defined Radio. 4

SKE Symmetric Key Encryption. 43, 49

SNR Signal-to-Noise Ratio. 9–12, 15, 52, 90, 105, 106, 109–112, 114, 116, 118, 129, 151

SPRT Weighted Sequential Probability Ratio Test. 37, 40, 41

SRTT Smoothed Round-Trip Time. 64

SSL Secure Sockets Layer. 29

ssthresh slow start threshold. 63

SYN Synchronization. 32

TCP Transmission Control Protocol. iii, vi, 18, 30, 32, 33, 58, 61–68, 70, 71, 73–85, 150, 151

TDD Time Division Duplex. 14

TDoA Time Difference of Arrival. 19, 54, 59, 88, 90, 91, 93–96, 98, 101–107, 111, 112, 114, 120, 122, 123, 126, 128, 129, 132, 136, 144, 151

TLS Transport Layer Security. 29

ToA Time of Arrival. 54, 88–91, 93, 94, 96, 105

TPC Trusted Platform Module. 16

TV Television. 4, 13–15, 36, 37, 50, 56, 59, 66, 75, 76, 87, 101, 102, 111, 118, 119, 121, 122, 149, 151, 152

twnd transmission window. 62, 63

UHF Ultra High Frequency. 76

UL Uplink. 14, 120, 121

VHF Very High Frequency. 76

W Watts. 13, 76

WEP Wired Equivalent Privacy. 29

WG Working Group. 13, 38, 83

WiMAX Worldwide Interoperability for Microwave Access. 13, 19

WLS Weighted Least Squares. 97, 105, 107, 122, 132, 133, 137, 141, 144, 145

WRAN Wireless Regional Area Networks. 13, 15, 16, 89, 101, 102, 111, 122

WSN Wireless Sensor Network. 51

WSPRT Weighted Sequential Probability Ratio Test. 41, 42

ZWP Zero Window Probe. 80

Bibliography

[802.22 WG, 2011] 802.22 WG (2011). IEEE standard for information technology–telecommunications and information exchange between systems wireless regional area networks (wran)–specific requirements part 22: Cognitive wireless ran medium access control (mac) and physical layer (phy) specifications: Policies and procedures for operation in the tv bands. IEEE Std 802.22-2011.

[Aad et al., 2004] Aad, I., Hubaux, J.-P., and Knightly, E. W. (2004). Denial of service resilience in ad hoc networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 202–215, New York, NY, USA. ACM.

[Aatique, 1997] Aatique, M. (1997). *Evaluation of TDOA techniques for position location in CDMA systems*. PhD thesis, Citeseer.

[Afolabi et al., 2009] Afolabi, O., Kim, K., and Ahmad, A. (2009). On secure spectrum sensing in cognitive radio networks using emitters electromagnetic signature. In *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on*, pages 1–5. IEEE.

[Akyildiz et al., 2006] Akyildiz, I. F., Lee, W.-Y., Vuran, M. C., and Mohanty, S. (2006). Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey. *Comput. Netw.*, 50(13):2127–2159.

[Akyildiz et al., 2011] Akyildiz, I. F., Lo, B. F., and Balakrishnan, R. (2011). Cooperative spectrum sensing in cognitive radio networks: A survey. *Physical Communication*, 4(1):40 – 62.

[Allman et al., 1999] Allman, M., Paxson, V., and Stevens, W. (1999). RFC 2581: TCP congestion control.

[Anand et al., 2008] Anand, S., Jin, Z., and Subbalakshmi, K. (2008). An analytical model for primary user emulation attacks in cognitive radio networks. In *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on*, pages 1–6. IEEE.

[Aso et al., 2001] Aso, M., Kawabata, M., and Hattori, T. (2001). A new location estimation method based on maximum likelihood function in cellular systems. In *Vehicular Technology Conference, 2001. VTC 2001 Fall. IEEE VTS 54th*, volume 1, pages 106–110. IEEE.

[Axelsson, 2000] Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy. Technical report, Department of Computer Engineering, Chalmers University of Technology, Göteborg, Sweden.

[Bace, 2000] Bace, R. G. (2000). *Intrusion Detection.* Sams.

[Bakre and Badrinath, 1994] Bakre, A. and Badrinath, B. (1994). I-tcp: Indirect tcp for mobile hosts. In *Distributed Computing Systems, 1995., Proceedings of the 15th International Conference on*, pages 136–143. IEEE.

[Bakshi et al., 1997] Bakshi, B., Krishna, P., Vaidya, N., and Pradhan, D. (1997). Improving performance of tcp over wireless networks. In *Distributed Computing Systems, 1997., Proceedings of the 17th International Conference on*, pages 365–373. IEEE.

[Balakrishnan et al., 1997] Balakrishnan, H., Padmanabhan, V., Seshan, S., and Katz, R. (1997). A comparison of mechanisms for improving tcp performance over wireless links. *Networking, IEEE/ACM Transactions on*, 5(6):756–769.

[Baldini et al., 2011] Baldini, G., Sturman, T., Biswas, A., Leschhorn, R., Godor, G., and Street, M. (2011). Security aspects in software defined radio and cognitive radio networks: A survey and a way ahead. *Communications Surveys Tutorials, IEEE*, PP(99):1 –25.

[Bhuse and Gupta, 2006] Bhuse, V. and Gupta, A. (2006). Anomaly intrusion detection in wireless sensor networks. *Journal of High Speed Networks*, 15(1):33–51.

[Blanton and Allman, 2002] Blanton, E. and Allman, M. (2002). On making tcp more robust to packet reordering. *SIGCOMM Comput. Commun. Rev.*, 32(1):20–30.

[Brennan, 2003] Brennan, D. (2003). Linear diversity combining techniques. *Proceedings of the IEEE*, 91(2):331 – 356.

[Brik et al., 2005] Brik, V., Rozner, E., Banerjee, S., and Bahl, P. (2005). Dsap: a protocol for coordinated spectrum access. In *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pages 611–614.

[Brown and Hwang, 1997] Brown, R. and Hwang, P. (1997). *Introduction to random signals and applied Kalman filtering*, volume 2. John Wiley & Sons.

[Brown and Sethi, 2008] Brown, T. and Sethi, A. (2008). Potential cognitive radio denial-of-service vulnerabilities and protection countermeasures: a multi-dimensional analysis and assessment. *Mobile Networks and Applications*, 13(5):516–532.

[Bucher and Misra, 2002] Bucher, R. and Misra, D. (2002). A synthesizable vhdl model of the exact solution for three-dimensional hyperbolic positioning system. *Vlsi Design*, 15(2):507–520.

[Buchwald et al., 2008] Buchwald, G., Kuffner, S., Ecklund, L., Brown, M., and Callaway, E. (2008). The design and operation of the ieee 802.22. 1 disabling beacon for the protection of tv whitespace incumbents. In *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on*, pages 1–6. IEEE.

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

[Burbank, 2008] Burbank, J. (2008). Security in cognitive radio networks: The required evolution in approaches to wireless network security. In *3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom)*, pages 1–7.

[Cabric et al., 2004] Cabric, D., Mishra, S., and Brodersen, R. (2004). Implementation issues in spectrum sensing for cognitive radios. In *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 772–776.

[Chen et al., 2009a] Chen, H., Jin, X., and Xie, L. (2009a). Reputation-based collaborative spectrum sensing algorithm in cognitive radio networks. In *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, pages 582–587. IEEE.

[Chen et al., 2008a] Chen, R., Park, J., and Bian, K. (2008a). Robust distributed spectrum sensing in cognitive radio networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 1876–1884. IEEE.

[Chen et al., 2008b] Chen, R., Park, J., Hou, Y., and Reed, J. (2008b). Toward secure distributed spectrum sensing in cognitive radio networks. *Communications Magazine, IEEE*, 46(4):50–55.

[Chen and Park, 2006] Chen, R. and Park, J.-M. (2006). Ensuring trustworthy spectrum sensing in cognitive radio networks. In *1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks (SDR)*, pages 110–119.

[Chen et al., 2008c] Chen, R., Park, J.-M., and Reed, J. (2008c). Defense against primary user emulation attacks in cognitive radio networks. *Selected Areas in Communications, IEEE Journal on*, 26(1):25–37.

[Chen et al., 2011] Chen, S., Zeng, K., and Mohapatra, P. (2011). Hearing is believing: Detecting mobile primary user emulation attack in white space. In *INFOCOM, 2011 Proceedings IEEE*, pages 36–40. IEEE.

[Chen et al., 2009b] Chen, Z., Cooklev, T., Chen, C., and Pomalaza-Raez, C. (2009b). Modeling primary user emulation attacks and defenses in cognitive radio networks. In *Performance Computing and Communications Conference (IPCCC), 2009 IEEE 28th International*, pages 208 –215.

[Chouinard et al., 2006] Chouinard, G., Cabric, D., and Gosh, M. (2006). IEEE P802.22 Wireless RANs - Sensing Thresholds.

[Clancy et al., 2007] Clancy, C., Hecker, J., Stuntebeck, E., and O'Shea, T. (2007). Applications of machine learning to cognitive radio networks. *Wireless Communications, IEEE*, 14(4):47–52.

[Clancy and Goergen, 2008] Clancy, T. and Goergen, N. (2008). Security in cognitive radio networks: Threats and mitigation. In *3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom)*, pages 1–8.

[Clancy and Khawar, 2009] Clancy, T. and Khawar, A. (2009). Security threats to signal classifiers using self-organizing maps. In *Cognitive Radio Oriented Wireless Networks and Communications, 2009. CROWNCOM'09. 4th International Conference on*, pages 1–6. IEEE.

[Clancy et al., 2011] Clancy, T., Khawar, A., and Newman, T. (2011). Robust signal classification using unsupervised learning. *Wireless Communications, IEEE Transactions on*, 10(4):1289–1299.

[Cordeiro et al., 2006] Cordeiro, C., Challapali, K., Birru, D., and Shankar, Sai, N. (2006). Ieee 802.22: an introduction to the first wireless standard based on cognitive radios. *Journal of Communications*, 1(1):38–47.

[Crohas, 2008] Crohas, A. (2008). Practical implementation of a cognitive radio system for dynamic spectrum access. Available at http://www.nd.edu/ jnl/pubs/crohas-ms-nd-2008.pdf.

[Digham et al., 2003] Digham, F., Alouini, M., and Simon, M. (2003). On the energy detection of unknown signals over fading channels. In *Communications, 2003. ICC'03. IEEE International Conference on*, volume 5, pages 3575–3579. Ieee.

[Digham et al., 2007] Digham, F. F., Alouini, M.-S., and Simon, M. K. (2007). On the energy detection of unknown signals over fading channels. *Communications, IEEE Transactions on*, 55(1):21 –24.

[Eksim et al., 2009] Eksim, A., Kulac, S., and Sazli, M. (2009). Effective cooperative spectrum sensing in ieee 802.22 standard with time diversity. In *International Conference on Advances in Computational Tools for Engineering Applications, ACTEA'09.*, pages 528–531.

[FCC, 2003a] FCC (2003a). Federal Communications Commission. ET Docket no. 03-122. Revision of Parts 2 and 15 of the Commissions Rules to Permit Unlicensed National Information Infrastructure (UNII) Devices in the 5GHz band.

[FCC, 2003b] FCC (2003b). Federal Communications Commission. ET Docket No. 03-322. Notice of proposed rule making and order.

[Floyd and Henderson, 1999] Floyd, S. and Henderson, T. (1999). Rfc 2582. the newreno modification to tcp's fast recovery algorithm.

[Foy, 1976] Foy, W. (1976). Position-location solutions by Taylor-series estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 12(2):187–194.

[Gezici et al., 2005] Gezici, S., Tian, Z., Giannakis, G., Kobayashi, H., Molisch, A., Poor, H., and Sahinoglu, Z. (2005). Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *Signal Processing Magazine, IEEE*, 22(4):70–84.

[Giannetsos et al., 2009] Giannetsos, T., Krontiris, I., Dimitriou, T., and Freiling, F. (2009). *On Security in RFID and Sensor Networks*, chapter Intrusion Detection in Wireless Sensor Networks. Auerbach Publications, CRC Press.

[Goff et al., 2000] Goff, T., Moronski, J., Phatak, D., and Gupta, V. (2000). Freeze-tcp: a true end-to-end tcp enhancement mechanism for mobile environments. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1537–1545.

[Greenstein et al., 1997] Greenstein, L., Erceg, V., Yeh, Y., and Clark, M. (1997). A new path-gain/delay-spread propagation model for digital cellular channels. *Vehicular Technology, IEEE Transactions on*, 46(2):477–485.

[He et al., 2010] He, A., Bae, K., Newman, T., Gaeddert, J., Kim, K., Menon, R., Morales-Tirado, L., Neel, J., Zhao, Y., Reed, J., et al. (2010). A survey of artificial intelligence for cognitive radios. *Vehicular Technology, IEEE Transactions on*, 59(4):1578–1592.

[Herath et al., 2011] Herath, S., Rajatheva, N., and Tellambura, C. (2011). Energy detection of unknown signals in fading and diversity reception. *Communications, IEEE Transactions on*, 59(9):2443 –2453.

[Hernández-Serrano et al., 2010] Hernández-Serrano, J., León, O., and Soriano, M. (2010). Modeling the Lion Attack in Cognitive Radio Networks. *EURASIP Journal on Wireless Communications and Networking*, 2011.

[Hernández-Serrano et al., 2011] Hernández-Serrano, J., León, O., and Soriano, M. (2011). Robust localization in WiMAX in the presence of malicious anchor nodes . *Submitted to Computers and Electrical Engineering.*

[Hernández-Serrano et al., 2008] Hernández-Serrano, J., Pegueroles, J., and Soriano, M. (2008). Shared self-organized GKM protocol for MANETs. *Journal of Information Science and Engineering (JISE)*, 24:1629–1646.

[Honkavirta et al., 2009] Honkavirta, V., Perala, T., Ali-Loytty, S., and Piche, R. (2009). A comparative survey of wlan location fingerprinting methods. In *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on*, pages 243–251. IEEE.

[Hoven and Sahai, 2005] Hoven, N. and Sahai, A. (2005). Power scaling for cognitive radio. In *Wireless networks, communications and mobile computing, 2005 International Conference on*, volume 1, pages 250–255. Ieee.

[Jacobson, 1988] Jacobson, V. (1988). Congestion avoidance and control. In *SIGCOMM '88: Symposium proceedings on Communications architectures and protocols*, pages 314–329, New York, NY, USA. ACM.

[Jin et al., 2009] Jin, Z., Anand, S., and Subbalakshmi, K. P. (2009). Mitigating primary user emulation attacks in dynamic spectrum access networks using hypothesis testing. *SIGMOBILE Mob. Comput. Commun. Rev.*, 13:74–85.

[Kaligineedi et al., 2008] Kaligineedi, P., Khabbazian, M., and Bhargava, V. (2008). Secure cooperative sensing techniques for cognitive radio systems. In *Communications, 2008. ICC'08. IEEE International Conference on*, pages 3406–3410. IEEE.

[Kaligineedi et al., 2010] Kaligineedi, P., Khabbazian, M., and Bhargava, V. (2010). Malicious user detection in a cognitive radio cooperative sensing system. *Wireless Communications, IEEE Transactions on*, 9(8):2488–2497.

[Kalman and Bucy, 1961] Kalman, R. and Bucy, R. (1961). New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83(3):95–108.

[Kalman et al., 1960] Kalman, R. et al. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45.

[Knapp and Carter, 1976] Knapp, C. and Carter, G. (1976). The generalized correlation method for estimation of time delay. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 24(4):320 – 327.

[Kuzmanovic and Knightly, 2006] Kuzmanovic, A. and Knightly, E. W. (2006). Low-rate tcp-targeted denial of service attacks and counter strategies. *IEEE/ACM Transactions on Networking (TON)*, 14(4):683–696.

[Lee et al., 1999] Lee, W., Stolfo, S., and Mok, K. (1999). A data mining framework for building intrusion detection models. In *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, pages 120–132.

[Lei and Chin, 2008] Lei, Z. and Chin, F. (2008). A reliable and power efficient beacon structure for cognitive radio systems. *Broadcasting, IEEE Transactions on*, 54(2):182–187.

[León et al., 2009a] León, O., Hernández Serrano, J., and Soriano, M. (2009a). A new cross-layer attack to tcp in cognitive radio networks. In *Second International Workshop on Cross Layer Design (IWCLD)*, pages 1–5.

[León et al., 2009b] León, O., Hernández-Serrano, J., and Soriano, M. (2009b). Un nuevo ataque a tcp para redes de radios cognitivas. In *Actas de las VIII Jornadas de Ingeniería Telemática (JITEL 2009)*, pages 83–90.

[León et al., 2010] León, O., Hernández-Serrano, J., and Soriano, M. (2010). Securing cognitive radio networks. *International Journal of Communication Systems*, 23(5):633–652.

[León et al., 2011] León, O., Hernández-Serrano, J., and Soriano, M. (2011). Cooperative detection of primary user emulation attacks in cognitive radio networks. In *(submitted to Computer Networks (Comnet))*.

[León et al., 2011a] León, O., Hernández-Serrano, J., and Soriano, M. (2011a). Robust Detection of Primary User Emulation Attacks in IEEE 802.22 Networks. In *4th International Conference on Cognitive Radio and Advanced Spectrum Management (CogArt 2011)*. ACM digital library.

[León et al., 2011b] León, O., Román, R., and Hernández-Serrano, J. (2011b). Towards a cooperative intrusion detection system for cognitive radio networks. In *NETWORKING 2011 Workshops*, pages 231–242. Springer.

[Li and Han, 2010] Li, H. and Han, Z. (2010). Dogfight in spectrum: Combating primary user emulation attacks in cognitive radio systems, part i: Known channel statistics. *Wireless Communications, IEEE Transactions on*, 9(11):3566 –3577.

[Li et al., 2007] Li, M., Koutsopoulos, I., and Poovendran, R. (2007). Optimal jamming attacks and network defense policies in wireless sensor networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1307 –1315.

[Li and Cadeau, 2011] Li, X. and Cadeau, W. (2011). Anti-jamming performance of cognitive radio networks. In *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*, pages 1–6. IEEE.

[Li et al., 2005] Li, Z., Trappe, W., Zhang, Y., and Nath, B. (2005). Robust statistical methods for securing wireless localization in sensor networks. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 91–98. IEEE.

[Littman, 1994] Littman, M. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the eleventh international conference on machine learning*, volume 157163. Citeseer.

[Liu et al., 2007] Liu, C., Scott, T., Wu, K., and Hoffman, D. (2007). Range-free sensor localization with ring overlapping based on comparison of received signal strength indicator. *Int. J. Sen. Netw.*, 2:399–413.

[Liu et al., 2005] Liu, D., Ning, P., and Du, W. (2005). Attack-resistant location estimation in sensor networks. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, pages 13–es. IEEE Press.

[Ludwig and Katz, 2000] Ludwig, R. and Katz, R. (2000). The eifel algorithm: making tcp robust against spurious retransmissions. *Computer Communication Review*, 30(1):30–36.

[Marias et al., 2006] Marias, G., Georgiadis, P., Flitzanis, D., and Mandalas, K. (2006). Cooperation enforcement schemes for manets: a survey. *Wireless Communications and Mobile Computing*, 6(3):319–332.

[Mascolo et al., 2001] Mascolo, S., Casetti, C., Gerla, M., Sanadidi, M., and Wang, R. (2001). Tcp westwood: Bandwidth estimation for enhanced transport over wireless links. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 287–297. ACM.

[Mathis et al., 1996] Mathis, M., Mahdavi, J., Floyd, S., and Romanow, A. (1996). Rfc 2018: Tcp selective acknowledgment options.

[Mathur and Subbalakshmi, 2007] Mathur, C. N. and Subbalakshmi, K. P. (2007). *COGNITIVE NETWORKS: Towards Self-Aware Networks*, chapter Security Issues in Cognitive Radio Networks, pages 293–284. John Wiley & Sons.

[MATLAB, 2009] MATLAB (2009). version 7.9 (r2009b).

[McHenry, 2005] McHenry, M. (2005). NSF spectrum occupancy measurements project summary.

[Mejia et al., 2009] Mejia, M., Pena, N., Munoz, J. L., and Esparza, O. (2009). A review of trust modeling in ad hoc networks. *Internet Research*, 19(1):88–104.

[Michiardi and Molva, 2003] Michiardi, P. and Molva, R. (2003). A game theoretical approach to evaluate cooperation enforcement mechanisms in mobile ad hoc networks. In *Proceedings of WiOpt*, volume 3. Citeseer.

[Milan et al., 2006] Milan, F., Jaramillo, J., and Srikant, R. (2006). Achieving cooperation in multihop wireless networks of selfish nodes. In *Proceeding from the 2006 workshop on Game theory for communications and networks*, pages 3–es. ACM.

[Min et al., 2010] Min, A., Shin, K., and Hu, X. (2010). Secure cooperative sensing in ieee 802.22 wrans using shadow fading correlation. *IEEE Transactions on Mobile Computing.*

[Mishra et al., 2004] Mishra, A., Nadkarni, K., and Patcha, A. (2004). Intrusion detection in wireless ad hoc networks. *Wireless Communications, IEEE*, 11(1):48–60.

[Mishra, 2004] Mishra, S. M. (2004). Corvus: A cognitive radio approach for usage of virtual unlicensed spectrum.

[Mitola, 2000] Mitola, J., I. (2000). *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio.* PhD thesis, Royal Institute of Technology (KTH), Sweeden.

[Nieminen et al., 2010] Nieminen, J., Jantti, R., and Qian, L. (2010). Primary User Detection in Distributed Cognitive Radio Networks under Timing Inaccuracy. In *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, pages 1–8. IEEE.

[Pahlavan et al., 2000] Pahlavan, K., Li, X., Ylianttila, M., Chana, R., and Latva-aho, M. (2000). An overview of wireless indoor geolocation techniques and systems. *Mobile and Wireless Communications Networks*, pages 1–13.

[Pandey and Agrawal, 2006] Pandey, S. and Agrawal, P. (2006). A survey on localization techniques for wireless networks. *Chinese Institute of Engineers*, 29(7):1125.

[Patwari et al., 2005] Patwari, N., Ash, J., Kyperountas, S., Hero, A.O., I., Moses, R., and Correal, N. (2005). Locating the nodes: cooperative localization in wireless sensor networks. *Signal Processing Magazine, IEEE*, 22(4):54–69.

[Paxson et al., 2000] Paxson, V., Allman, M., and Timer, C. (2000). Rfc 2988. computing tcp's retransmission timer.

[Peng et al., 2009] Peng, Q., Cosman, P., and Milstein, L. (2009). Tradeoff between spoofing and jamming a cognitive radio. In *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*, pages 25–29. IEEE.

[Peng et al., 2011] Peng, Q., Cosman, P., and Milstein, L. (2011). Spoofing or jamming: Performance analysis of a tactical cognitive radio adversary. *Selected Areas in Communications, IEEE Journal on*, 29(4):903–911.

[Priyantha et al., 2000] Priyantha, N. B., Chakraborty, A., and Balakrishnan, H. (2000). The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 32–43, New York, NY, USA. ACM.

[Psaras and Tsaoussidis, 2007] Psaras, I. and Tsaoussidis, V. (2007). The tcp minimum rto revisited. *NETWORKING 2007. Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, pages 981–991.

[Puterman, 1994] Puterman, M. (1994). *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons, Inc.

[Qin et al., 2009] Qin, T., Yu, H., Leung, C., Shen, Z., and Miao, C. (2009). Towards a trust aware cognitive radio architecture. *ACM SIGMOBILE Mobile Computing and Communications Review*, 13(2):86–95.

[Qiu et al., 2006] Qiu, L., Yang, Y. R., Zhang, Y., and Shenker, S. (2006). On selfish routing in internet-like environments. *IEEE/ACM Transactions on Networking (TON)*, 14(4):725–738.

[Rong and Sichitiu, 2006] Rong, P. and Sichitiu, M. (2006). Angle of arrival localization for wireless sensor networks. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, volume 1, pages 374 – 382.

[Roughgarden and Tardos, 2002] Roughgarden, T. and Tardos, E. (2002). How bad is selfish routing? *Journal of the ACM (JACM)*, 49(2):236–259.

[Rousseeuw et al., 1987] Rousseeuw, P., Leroy, A., and Wiley, J. (1987). *Robust regression and outlier detection*, volume 3. Wiley Online Library.

[Russell and Norvig, 2002] Russell, S. and Norvig, P. (2002). *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2nd edition.

[Savvides et al., 2001] Savvides, A., Han, C., and Strivastava, M. (2001). Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179. ACM.

[Sawada et al., 2006] Sawada, M., Cossette, D., Wellar, B., and Kurt, T. (2006). Analysis of the urban/rural broadband divide in canada: Using gis in planning terrestrial wireless deployment. *Government Information Quarterly*, 23(3-4):454–479.

[Security and Council, 2004] Security, M. and Council, R. (2004). Communications infrastructure security, access, and restoration. final report.

[Simic and Sastry, 2001] Simic, S. and Sastry, S. (2001). Distributed localization in wireless ad hoc networks. *UC Berkeley ERL report*.

[Song and Zhang, 2009] Song, C. and Zhang, Q. (2009). Achieving cooperative spectrum sensing in wireless cognitive radio networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 13(2):14–25.

[Srinivasan and Wu, 2007] Srinivasan, A. and Wu, J. (2007). A survey on secure localization in wireless sensor networks. *Encyclopedia of Wireless and Mobile communications*.

[Tech, 2011] Tech, W. V. (2011). Ossie:open source software defined radio. http://ossie.wireless.vt.edu.

[Tian et al., 2005] Tian, Y., Xu, K., and Ansari, N. (2005). Tcp in wireless environments: problems and solutions. *Communications Magazine, IEEE*, 43(3):S27–S32.

[Tkachenko et al., 2006] Tkachenko, A., Cabric, D., and Brodersen, R. (2006). Cognitive radio experiments using reconfigurable BEE2. In *Signals, Systems and Computers, 2006. ACSSC'06. Fortieth Asilomar Conference on*, pages 2041–2045. IEEE.

[Toonstra and Kinsner, 1996] Toonstra, J. and Kinsner, W. (1996). A radio transmitter fingerprinting system odo-1. In *Canadian Conference on Electrical and Computer Engineering*, volume 1, pages 60–63.

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

[USC/ISI et al., 2007] USC/ISI, PARC, X., UCB, SAMAN, CONCER, ACIRI, and etc. (2007). The Network Simulator - ns-2 2.31 release. http://www.isi.edu/nsnam/ns/.

[Visotsky et al., 2005] Visotsky, E., Kuffner, S., and Peterson, R. (2005). On collaborative detection of TV transmissions in support of dynamic spectrum sharing. In *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pages 338–345. IEEE.

[Walker, 2000] Walker, J. (2000). IEEE 802.11 Wireless LANs: unsafe at any key size; an analysis of the WEP encapsulation. Technical Report IEEE 802.11-00/362, Intel Corporation.

[Wallner et al., 1998] Wallner, D., Harder, E., and Agee, R. (1998). Key management for multicast: issues and architectures. RFC 2627.

[Wang et al., 2011] Wang, B., Wu, Y., Liu, K., and Clancy, T. (2011). An anti-jamming stochastic game for cognitive radio networks. *Selected Areas in Communications, IEEE Journal on*, 29(4):877–889.

[Wang and Zhang, 2002] Wang, F. and Zhang, Y. (2002). Improving tcp performance over mobile ad-hoc networks with out-of-order detection and response. In *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 217–225, New York, NY, USA. ACM.

[Wang et al., 2006] Wang, Y., Attebury, G., and Ramamurthy, B. (2006). A survey of security issues in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 8(2):2–23.

[Wei et al., 2006] Wei, D. X., Jin, C., Low, S. H., and Hegde, S. (2006). Fast tcp: motivation, architecture, algorithms, performance. *IEEE/ACM Trans. Netw.*, 14(6):1246–1259.

[Wu et al., 2010] Wu, Y., Wang, B., and Liu, K. (2010). Optimal defense against jamming attacks in cognitive radio networks using the markov decision process approach. In *IEEE Globecom 2010 proceedings.*

[Xu et al., 2004] Xu, K., Tian, Y., and Ansari, N. (2004). Tcp-jersey for wireless ip communications. *Selected Areas in Communications, IEEE Journal on*, 22(4):747–756.

[Xu et al., 2007] Xu, W., Trappe, W., and Zhang, Y. (2007). Channel surfing: defending wireless sensor networks from interference. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 499–508. ACM.

[Yu et al., 2009] Yu, K., Sharp, I., and Guo, Y. (2009). *Ground-based wireless positioning*, volume 5. Wiley-IEEE Press.

[Yuan et al., 2007] Yuan, Y., Bahl, P., Chandra, R., Chou, P., Ferrell, J., Moscibroda, T., Narlanka, S., and Wu, Y. (2007). Knows: Cognitive radio networks over white spaces. *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, pages 416–427.

[Zeng et al., 2010] Zeng, Y., Liang, Y.-C., Hoang, A. T., and Zhang, R. (2010). A review on spectrum sensing for cognitive radio: challenges and solutions. *EURASIP J. Adv. Signal Process*, 2010:2:2–2:2.

[Zhang and Lee, 2000] Zhang, Y. and Lee, W. (2000). Intrusion detection in wireless ad-hoc networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 275–283, New York, NY, USA. ACM.

[Zhang et al., 2003] Zhang, Y., Lee, W., and Huang, Y. (2003). Intrusion detection techniques for mobile wireless networks. *Wireless Networks*, 9(5):545–556.

[Zhang and Xie, 2007] Zhang, Z. and Xie, X. (2007). Intelligent cognitive radio: Research on learning and evaluation of CR based on Neural Network. In *Information and Communications Technology, 2007. ICICT 2007. ITI 5th International Conference on*, pages 33–37. IEEE.

[Zhengyi et al., 2010] Zhengyi, L., Lin, L., and Chi, Z. (2010). Fast Detection Method in Cooperative Cognitive Radio Networks. *International Journal of Digital Multimedia Broadcasting*, 2010.