*Journal of Ubiquitous Systems & Pervasive Networks*
*Volume 5, No. 1 (2014) pp. 01-08*

# Detecting Primary User Emulation Attacks in Cognitive Radio Networks via Physical Layer Network Coding

**Xiongwei Xie[a], Weichao Wang[a]***

[a]*Department of SIS, UNC Charlotte, Charlotte, NC, 28223, USA*

## Abstract

Primary user emulation (PUE) attacks on cognitive radio networks pose a serious threat to the deployment of this technique. Previous approaches usually depend on individual or combined received signal strength (RSS) measurements to detect emulators. In this paper, we propose a new mechanism based on physical layer network coding to detect the emulators. When two signal sequences interfere at the receiver, the starting point of collision is determined by the distances among the receiver and the senders. Using the signal interference results at multiple receivers and the positions of reference senders, we can determine the position of the 'claimed' primary user. We can then compare this localization result with the known position of the primary user to detect the PUE attack. We design a PUE detection mechanism for wireless networks with trustworthy reference senders. We analyze the overhead of the proposed approach and study its detection accuracy through simulation.

***Keywords:*** *Cognitive Radio Networks, Primary User Emulation, Physical Layer Network Coding.*

## 1. Introduction

Many researchers, including the former US FCC (Federal Communications Commission) Chairman Julius Genachowski, believe that there is a looming spectrum crisis at the frequencies that can be economically used for wireless communications. This point is strengthened when we look at the multiple allocations over all of the frequency bands in the FCC frequency chart. This has resulted in a tight competition for the use of spectra, especially in the bands below 3 GHz. On the other hand, a large portion of the assigned spectrum is used sporadically. For example, in Figure 1 we find that the spectrum usage is concentrated on certain portions of the spectrum while a significant amount of the spectrum remains unutilized. This appears to be a contradiction to the concern of spectrum shortage since in fact we have an abundant amount of spectrum.

To solve this problem, dynamic spectrum access is proposed. The dynamic spectrum access technique allows wireless nodes to use spectrum sensing to identify the 'white spaces' in licensed spectrum. The cognitive radios will then opportunistically utilize these white spaces. To avoid any interference with the primary users, a secondary user must leave the occupied channels if it detects a primary user. Therefore, one of the major technical challenges in spectrum sensing is the problem of precisely identifying the signals of the real primary users. The malicious secondary users can mimic the spectral characteristics of primary users to gain priority access to the wireless channels, which is called "primary user emulation" (PUE) attacks.

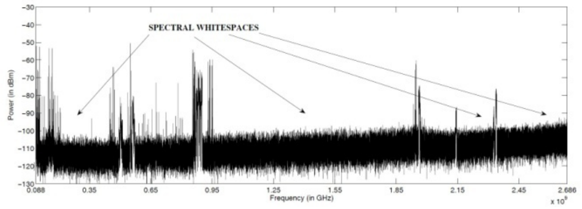

**Fig.1. A snapshot of power spectral density from 88 MHz to 2686 MHz measured in Worcester, Massachusetts, US.**

From a security perspective, the PUE attack can be viewed as an authentication problem. However, the traditional authentication mechanisms based upon the cryptographic signatures cannot be directly applied since the FCC states very clear that "no modification to the incumbent system (i.e., primary user) should be required" [1]. Therefore, other schemes of authentication must be designed to defend against such attacks.

Existing approaches to detecting the PUE attacks can be divided into two groups: communication oriented and localization oriented. In the first group, the secondary nodes use the spectrum sensing techniques to match the characteristics of the radio signals to those of the primary user. The detection mechanisms include filter and cyclostationary feature detection [2], spectrum decision and channel

[*] Corresponding author. Tel.: +17046877987
Fax: +17046874893; E-mail: weichaowang@uncc.edu

DOI: 10.5383/JUSPN.05.01.001

parameters [3], shadow senders [4], and static helper nodes [5]. In the second group, the researchers use the received signals to estimate the position of the sender. They have designed different methods to model the communication channels and improve the signal measurement accuracy [6]. Outliers in localization procedures are filtered out to improve the detection accuracy of PUE attacks [7].

When we investigate the existing approaches, we find that several issues may impact their wide adoption. First, some approaches require the deployment of some special hardware [4, 8] or the adoption of complex software [9] to achieve attack detection. These approaches will cause an increased deployment cost. Second, the power level of the received signals can be impacted by many factors and could have fluctuated in a wide range. The attackers can take advantage of this property to impersonate the real primary user. Therefore, a new mechanism to detect the PUE attacks is needed to solve these problems.

In this paper, we propose a PUE attack detection mechanism based on the physical layer network coding (PNC) technique. PNC uses the additive nature of the electromagnetic waves to serve as the coding procedure. In our approach, we estimate the position of a wireless node by letting its radio signals interfere with a reference sender. These interfered sequences will be captured by multiple secondary users. Combining the starting points of signal interference results with their positions, the secondary users will determine a group of hyperbolas on which the wireless sender resides. Then they will compare the intersection point of these hyperbolas with the known position of the primary user to detect the PUE attack.

To turn the approach into a practical solution, research challenges from multiple aspects must be carefully addressed. From the network point of view, we need to verify the authenticity of the received signals and accurately locate the position of the sender. From the security point of view, we need to design mechanisms to identify the false claims of positions and signal interference results provided by malicious nodes.

Compared to previous approaches to PUE attack detection, our investigation has the following contributions: (1) The research will demonstrate that in addition to improving the bandwidth usage efficiency in wireless networks, physical layer network coding can also be used to detect malicious attacks. (2) The proposed PUE attack detection mechanism does not require the deployment of any special hardware. The assumed trustworthy reference senders already exist in the IEEE standards such as 802.22 and 802.16h. (3) The overhead and detection accuracy of the approach are studied through both theoretical analysis and simulation.

The remainder of this paper is organized as follows. In Section 2, we introduce the basic idea of using PNC to achieve localization of wireless nodes. In Section 3, we present the details of the proposed approach. The overhead and detection accuracy of the approach are studied in Section 4. Section 5 discusses several methods to improve the detection accuracy. Finally, Section 6 concludes the paper.

## 2. Localization through Physical Layer Network Coding

### 2.1. System Assumptions

In the investigated networks, we assume that the primary users, the secondary users, and the attackers all use omni-directional antennas. Extending the approach to directional antennas will be investigated in the future. Although the proposed approach can be applied to the systems with multiple primary users, in the following sections we will assume a single primary user. The primary user is located at a fixed position and both the secondary users and the attackers know its position. In real life applications, the coverage range of a primary user (e.g. a TV station) is usually much larger than that of the secondary users (e.g. a cognitive radio device). For example, a TV tower has the transmission power of hundreds of thousands of Watts. On the contrary, the secondary users or the attackers are usually normal cognitive radio devices that have a transmission power of tens of Watts. FCC requires all TV towers or radio stations to enforce strict physical security. Therefore, similar to [4], we assume that the secondary users or the attackers cannot be physically close to the primary user.

We use the disk graph model to describe the communication ranges of the secondary users and attackers. The signals from the secondary users can be correctly received by all nodes within the distance $r$. We assume that every secondary user learns its current position through the GPS chip set. The GPS chip will also provide loosely synchronized clocks to the users. We assume that the wireless nodes share a secure, lightweight pseudo random bit generator (PRBG) [10]. When an attacker or a legitimate secondary user sends out a packet, all receivers can authenticate the sender and verify the integrity of the packet. This can be achieved through the Message Authentication Code (MAC) embedded in the packet. The details of packet authentication will be discussed later.

We assume that an attacker can act as a legitimate secondary user and also has the resources such as the PRBG and GPS. An attacker has a total control over the signal sequences that it sends out and it can mimic a primary user's radio signal. The attacker can adjust its transmission power. Multiple attackers can collaborate to conduct a PUE attack. However, we assume that the attackers do not have the computation power to directly compromise the encryption keys of other legitimate users or reverse a secure hash function.

### 2.2. Use PNC to Achieve Node Localization

Figure 2 illustrates the basic idea of physical layer network coding. In the topology, $A$ and $C$ depend on $B$ to forward the frames between them. In the PNC approach, $A$ and $C$ will send out their packets and $B$ will receive the interference results of the two frames. It will rebroadcast the received signals to both $A$ and $C$ so that they can leverage their knowledge about $frame1$ and $frame2$ respectively to separate the signals and recover the data. Please note that the PNC based mechanism does not require the frames to reach the receiver simultaneously since it can accurately locate the starting point of signal collisions [11].

We can use PNC to calculate the position of a wireless node. We use $d_{MN}$ to represent the distance between two nodes $M$ and $N$. We use $T$ to represent a specific moment and $t$ to represent a time duration. If radio waves propagate at the speed $s$, the transmission delay between $M$ and $N$ will be $\frac{d_{MN}}{s}$. In our analysis, we measure the difference between the arriving time of two sequences based on the starting point of signal interference. We can locate the symbol in the sequence from which the collision starts. Then we can translate this information into a time difference based on the frequency of the radio signals.

Figure 2 also illustrates an example of radio signals colliding at wireless receivers. We assume that four nodes $A$, $C$, $D$, and $E$ can receive the signals from each other. We also assume that nodes *C*, *D*, and *E* know their positions. Node *A* wants to determine its position. Two anchor nodes *C* and *D* send out signal sequences that will collide at both *A* and *E*. Without losing generality, we assume that *C* starts sending at $T_C = 0$ and *D* starts sending at $T_D \geq 0$.

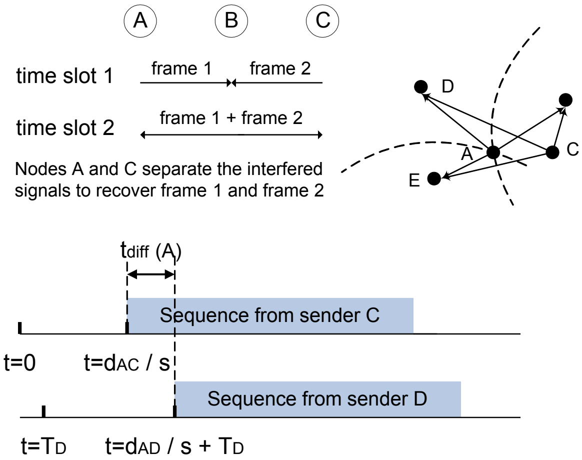

**Fig. 2. Node localization through physical layer network coding. Top left: physical layer network coding. Top right: node A is at the intersection of two hyperbolas. Bottom: difference between the arriving time of two sequences at node A.**

Node *A* will receive the sequence from *C* at $\frac{d_{AC}}{s}$, and the sequence from *D* at $(T_D + \frac{d_{AD}}{s})$. The difference between the arriving time is $t_{diffA}$= $(T_D+\frac{d_{AD}-d_{AC}}{s})$. Similarly, we can calculate the difference between the arriving time at node *E* as $t_{diffE}$= $(T_D+\frac{d_{ED}-d_{EC}}{s})$. The difference between $t_{diffA}$ and $t_{diffE}$ is:

$$t_{diffE} - t_{diffA} = \frac{d_{ED}-d_{EC}}{s} - \frac{d_{AD}-d_{AC}}{s} \qquad (1)$$

We simplify this equation and will get:

$$d_{AD} - d_{AC} = (d_{ED} - d_{EC}) + s \times (t_{diffA} - t_{diffE}) \qquad (2)$$

Since nodes *C*, *D*, and *E* know their positions, they can calculate $d_{ED} - d_{EC}$. Using the values of $t_{diffA}$ and $t_{diffE}$, we can calculate $d_{AD} - d_{AC}$. Since nodes *C* and *D* know their positions, node *A* will reside on one wing of the hyperbola that is jointly determined by the positions of *C* and *D* and the value of $d_{AD} - d_{AC}$. We need more hyperbolas to determine the position of node *A*. We can choose other pairs of anchor nodes to determine more hyperbolas. Node *A* will be positioned at the intersection point of these hyperbolas, as shown in Figure 2.

The localization procedure based on the PNC technique has several highly desirable properties. First, since the mechanism uses only the starting points of the collisions to determine the hyperbolas and calculate the position of the node, we do not need the wireless nodes to synchronize the arrival time of their data sequences. As illustrated in Equation (1), the parameter $T_D$ has been canceled out, which means that the difference between the transmission time of the two senders will not impact the proposed approach. Second, the proposed mechanism does not require the wireless nodes to be equipped with special hardware such as directional antennas or dual signal sources, which will result in a lower node cost and easier deployment. Third, the proposed approach works in a distributed manner and does not require a centralized controller. With these properties, the approach can be easily adopted by cognitive radio networks.

## 3. Detecting the PUE Attack

### 3.1. Overview of the Approach

In the remainder of the paper, we call the sender whose position we try to locate as the **interested sender** (it could be the primary user or an emulator), and the sender of the interfered signals as the **reference sender**. The PNC based localization technique provides a very promising approach to distinguishing the real primary user from an emulator: when an unknown signal is detected, a legitimate secondary user can intentionally send out a sequence to interfere with the signal. Other secondary users can capture the interference results and determine the hyperbolas. If the intersection point of the hyperbolas is at the known position of the primary user, the secondary users will leave the channel. Otherwise, they will stay there.

The major challenge that we face is the safety of the approach. Since we cannot distinguish an attacker from a legitimate secondary user, the attackers can participate in the localization procedure. They can send out false information about their positions and interference results to mislead the calculation procedures. Therefore, we must design mechanisms to defend against such attacks. In the following scenario, we assume that trustworthy reference senders exist in the network. This scenario matches the application environments of the IEEE 802.22 [12] and 802.16h [13] network standards. The trustworthy nodes can serve as the reference senders during PUE detection. We assume that the signals from a trustworthy sender *TR* can be correctly received by *p* legitimate secondary users $\{s_1, s_2, \cdots, s_p\}$ and *q* attackers $\{m_1, m_2, \cdots, m_q\}$. At the same time, we also assume that all these nodes can correctly receive the signals from the real primary user *P*.

When *TR* senses the communication channel and detects some signals that could have come from the real primary user, it will initiate the PUE detection procedure. *TR* will choose a random number as the seed for the PRBG to generate a random bit sequence and use the sequence to fill a data packet. When it sends out the packet, the radio waves from *TR* will interfere with the signals from the primary user (or an emulator). Message authentication codes (MAC) will be attached to the packets to protect their authenticity and integrity. The details of the MAC codes will be discussed later.

Using the mechanism described in [11, 14], the wireless nodes can detect the signal collision and record the interference results. Using the MAC code from *TR*, they can verify the identity of the sender and integrity of the information. They will then use the PRBG to regenerate the random sequence. Combining the interference results with the regenerated sequence, the receivers can recover the packet from the interested sender. The receivers can then calculate the $t_{diff}$ values based on the starting points of interference and the frequency of the radio signals. Now every receiver (both legitimate secondary users and attackers) will exchange its position, its $t_{diff}$ value, and the hash result of the recovered packet from the interested sender with its neighbors. The broadcast packets will be protected by the MAC codes so that the receivers can verify their contents. The secondary users can combine the $t_{diff}$ values with the node positions to determine the position of the interested sender. Once the position is determined, secondary users can compare it with the known position of the primary user to determine whether or not they are under a PUE attack.

Please note that it will be very difficult for an attacker to impersonate the trustworthy sender $TR$. Since the seed is protected by the MAC code, every legitimate secondary user will regenerate the correct sequence from $TR$. Now let us assume that the legitimate secondary users actually receive the signal interference results of the sequences from the primary user $P$ and an attacker. The secondary users can still subtract the sequence of $TR$ from the interference results. However, since the secondary users have different $t_{diff}$ values, every secondary user will have a different recovered sequence of the primary user. When they broadcast the hash results, the secondary users can easily detect the abnormal. As another attack, the malicious nodes can send out wrong hash results of the recovered sequence of the primary user. Since all of the legitimate secondary users have the same hash results, they can form a group to conduct the PUE attack detection and ignore whoever has a different hash value.

### 3.2. Construct a Practical Approach

**Who are the Trustworthy Senders**

One big concern of the proposed approach is which nodes can be used as the trustworthy senders. Fortunately, several IEEE standards using the cognitive radio (CR) technique such as 802.22 (CR for Wireless Regional Area Network) [12] and 802.16h (CR for WiMAX) [13] assume the existence of base stations. Many of these base stations are deployed by the cellular phone/network service providers. Therefore, these base stations can serve as the trustworthy senders. The standards such as 802.22 also require the base stations to have GPS devices and loosely synchronized clocks, which can be used for PNC based localization [15] and hash chain based authentication [16], respectively.

**Authentication of the Packets**

In Section 3.1 we have assumed that the wireless nodes can attach a MAC code to the packet to protect its authenticity and integrity. Although this problem can be solved by assigning a different public/private key pair to every node, the computation overhead of the digital signatures can be too heavy for the mobile devices. We propose to use the same method as in [4] to accomplish the task. We assume that every node can generate a random number $y_i$ and use a secure hash function to construct an $l$-entry one-way hash chain $hash^j(y_i)$, ($l \geq j \geq 0$). If you have the knowledge of $hash^{j_1}(y_i)$, it will be very easy for you to authenticate $hash^{j_2}(y_i)$, when we have $l \geq j_1 \geq j_2 \geq 0$. However, the one-way property will prevent an attacker from calculating an earlier entry in the hash chain.

With this observation, every node needs to sign only the last entry in the hash chain with its private key and distribute it to the neighbors. The receivers can verify the signatures and keep a record of the hash chains for the nodes. Then the nodes can use the entries in the hash chains in the reverse order to achieve packet authentication [17, 18]. Before an entry in the hash chain is disclosed, the knowledge of that entry can be used to authenticate the packet. After an entry is released, all other nodes can use it to regenerate the MAC code for authentication.

To prevent the malicious attackers from using the already disclosed hash chain entries to generate fake messages, the wireless nodes need loosely synchronized clocks to link the release of the hash entries to specific time points. In this way, the receivers can easily determine whether or not the MAC code is generated by the original owner. The accuracy of GPS clocks is good enough for hash chain release management [19].

**Detection of Collision**

The secondary users need to distinguish three states of the system: no signal, one incoming sequence, and two colliding sequences. To detect the arrival of the first data sequence, the receiver can monitor the incoming energy level since the received signal demonstrates a much higher energy level than that of the noises.

Since our approach does not require the wireless nodes to synchronize the arrival time of multiple sequences, there is a good chance that the two sequences will arrive at the receiver at different time points. Therefore, the receiver must be able to locate the starting point of the collision. Before this point, the receiver runs standard decoding. After this point, the receiver needs to separate the interfered signals. To distinguish the two states, the receiver needs to adopt different mechanisms based on the signal modulation schemes. Below we use the minimum-shift keying (MSK) modulation as an example to explain the procedure. MSK represents the data bits by varying the phase difference between consecutive complex signals. Specifically, a phase difference of π/2 represents bit ‘1’, and a difference of −π/2 represents bit ‘0’. The receiver can measure the variance in the energy level of the incoming signals. Since MSK encodes the bits in the phase, the energy of a non-interfered signal is almost constant. When two signals collide at the receiver, the variance will become much larger [11]. Therefore, we can set up a threshold. When the variance becomes larger than the value, the sequence separation algorithm will be executed.

**Detection of the Real Primary User**

Although in Section 3.1 we focus on the detection of the PUE attack, the same localization procedure can be used to detect the real primary user. When the reference sender sends out its real position, the detection procedure of the real primary user is exactly the same. Here all the hyperbolas determined by the legitimate secondary users will intersect at the primary user. The attackers, on the contrary, will use the false positions and $t_{diff}$ values to mislead the legitimate nodes. For the same network scenario, the false positive and false negative alarms will have exactly the same curve. The simulation results will be presented in Section 4.

### 3.3. Safety of the Approach

We assume that an emulator $U$ tries to impersonate the real primary user $P$. During the PUE detection procedure, for any legitimate secondary user $s_i$ ($i \in 1 \cdots p$), it will get the positions and the $t_{diff}$ values from $(p-1)$ legitimate secondary users and $q$ attackers. Since the received information is protected by the MAC codes of the senders, the attackers cannot impersonate other legitimate users. Using Equation (1), $s_i$ will alternatively combine its own information with information from the other $p+q-1$ nodes to determine $p+q-1$ independent hyperbolas. Since the position information and $t_{diff}$ values from the legitimate secondary users are true, the $p-1$ independent hyperbolas that are determined based on $s_i$ and $s_k$ ($k = 1 \cdots p, k \neq i$) will all pass through the position of node $U$.

To assist the emulator $U$ to defeat the detection procedure, the attackers have to lie about their positions and $t_{diff}$ values. Since information from the attackers contains their MAC codes, a single attacker cannot send different position and $t_{diff}$values to different legitimate secondary users. Now we assume that the real position and $t_{diff}$ value of the attacker $m_j$ are $Posi(m_j)$ and $t_{diff}(m_j)$, respectively. $m_j$ will send out the false information $Posi(\overline{m_j})$ and $t_{diff}(\overline{m_j})$ to the legitimate secondary users. In the following description, we use the overhead bar $\overline{m_j}$ to represent the values calculated based on the false information. For the legitimate secondary user $s_i$, to allow the hyperbola determined by $s_i$ and $m_j$ to pass through the position of the primary user $P$, $m_j$ must make the false values satisfy:

$$d_{s_ip} - d_{\overline{m_j}p} = \left(d_{s_iTR} - d_{\overline{m_j}TR}\right) + s \times (t_{diff}(s_i) - t_{diff}(\overline{m_j}))$$ (3)

As our previous analysis shows, $m_j$ must send the same $Posi(\overline{m_j})$ and $t_{diff}(\overline{m_j})$ to all legitimate secondary users. To fool as many legitimate nodes as possible, the attacker needs to solve the following problem: given the positions of the primary user $P$, the emulator $U$, and the legitimate nodes $s_i(i = 1 \cdots p)$, an attacker needs to calculate the fake information $Posi(\overline{m_j})$ and $t_{diff}(\overline{m_j})$ so that all hyperbolas determined by $m_j$ and $s_i$ $(i = 1 \cdots p)$ will pass through $P$.

This problem is similar to the GPS spoofing attack that is studied in [20]. In their approach, the authors study the relationship between the number of legitimate receivers to be fooled and possible positions of the satellite impersonator. The results are shown in Table 1. In our approach, since emulator $U$ is fixed, the satellite impersonator is replaced by the fake position information $Posi(\overline{m_j})$ of $m_j$. For example, if we want to mislead the localization results of four secondary users, $m_j$ must be positioned at one of the two points. From Table 1, we can see that it is almost impossible to satisfy the requirements listed above when there are more than three legitimate secondary users in the neighborhood since all malicious nodes will be located at those two points.

**Table 1. Relationship between number of victims and possible positions of the emulator**

| number of victims | possible positions of the emulator |
|---|---|
| 2 | Set of hyperboloids |
| 3 | Set of intersections of two hyperboloids |
| 4 | Set of two points |
| ≥5 | Set of specific points |

Based on the results in [20], we adopt the following scheme to determine the position of the interested sender. We will choose a threshold value *thresh*. For a secondary user $s_i$, only when there are at least *thresh* independent hyperbolas with $s_i$ as one of the focal points passing through the same point, it will be used as the position of the interested sender. If multiple positions satisfy this requirement, $s_i$ will choose the position with the largest number of hyperbolas as the interested sender. The legitimate node can make a random selection if multiple positions have the same number of hyperbolas.

**Node that cannot Reach the Threshold**

A secondary user needs at least *thresh* independent hyperbolas using it as a focal point to intersect at the same position to locate the interested sender. For various reasons (e.g. low node density), some legitimate secondary users may not be able to reach the threshold value. To solve this problem, one method can be adopted. The trustworthy sender can increase its transmission power so that more nodes can capture the interfered signals. The cost to this approach, however, is that the secondary users have to exchange the position information and $t_{diff}$ values with the nodes in a larger range. This will lead to extra communication overhead and power consumption at the users.

## 4. Analysis and Simulation

### 4.1. Overhead of the Proposed Approach

Since the proposed approach incurs very little storage overhead at the secondary users, our analysis will focus on the computation and communication overhead. The majority of the computation overhead is caused by solving the hyperbolas to determine their intersection points. Since a hyperbola can be represented as a second-degree equation in the Cartesian coordinates, determining the intersections of two hyperbolas can be viewed as a procedure to solve two second-degree equations. Several mechanisms to efficiently calculate the intersections of hyperbolas have been proposed [21, 22]. In [23] the authors propose a mechanism that uses only simple add and shift operations in the computation. Therefore, it can be easily implemented in hardware or firmware. Research has also shown that this method outperforms the traditional schemes in terms of the required number of operations for a specific accuracy level.

If there are $p$ legitimate secondary users and $q$ attackers, a legitimate secondary user $s_i$ will determine $(p + q - 1)$ independent hyperbolas. If we have to calculate the intersection point of every two hyperbolas, there will be $(p + q - 1)(p + q - 2)/2$ cases we need to solve, which could be a pretty large number. Fortunately, several schemes can be used to greatly reduce the computation overhead. First, all of the $(p - 1)$ hyperbolas determined by the legitimate users will have the same intersection point. In this way, we only need to solve the intersection point of two hyperbolas and then we can easily verify whether or not the point is on the other hyperbolas. If a majority of the wireless nodes are legitimate, say $\frac{p}{p+q} = 0.8$, this scheme can reduce about 64% of the computation overhead. In the second scheme, the secondary users can implement a simplified version of the proposed approach. Here the secondary user will just examine whether or not a hyperbola passes through the position of the primary user $P$ and count the total number. Since the position of $P$ is known to every secondary user, the computation overhead will be very low. The cost of this simplified implementation is that the secondary users will not know whether or not there is another joint intersection point of more than *thresh* hyperbolas. This may lead to the increase in false alarm rate.

The majority of the communication overhead of the proposed approach comes from the exchange of the positions, $t_{diff}$ values, hash results, and MAC codes. Every secondary user needs to send out its own information and receive $p + q - 1$ copies from other nodes. We assume that the packets sent out by the secondary users contain $l$ bytes. Therefore, all users need to send out at most $l \times (p + q - 1 + 1 + 1) = l \times (p + q + 1)$ bytes. If we assume that $l$ =128 Bytes, and $p + q$ has the value of 10, in every round of PUE detection the secondary users need to send out 1.4K Bytes altogether, which can be easily handled by modern wireless devices. The secondary users also need to capture the signal interference results from the primary user and the reference sender. Note that the length of the interference result is at most two times of the longest packet in the network.

### 4.2. Simulation Results

We assess the detection accuracy of the proposed approach through simulation. We assume a network area of 2000 × 2000 $m^2$. Both the legitimate secondary users and the attackers are randomly and uniformly distributed in the network [24]. The radio communication range is $250m$. We assume that the trustworthy senders are also randomly distributed in the network and every secondary user (both legitimate and malicious) is covered by at least one sender. We study the impacts of the legitimate user and attacker densities, and the selected threshold value on the detection accuracy. We focus on false negative alarms, in which an emulator is incorrectly identified as the primary user.

### *4.2.1. Selection of the threshold value*

As we describe in Section 3.3, only when at least *thresh* independent hyperbolas determined by a legitimate secondary user pass through the same point, that point will be used as the position of the interested sender. Therefore, the selection of the threshold value will directly impact the detection capability of the proposed approach. If *thresh* is too large, very few legitimate users will be able to collect enough information from the other nodes under the same trustworthy sender to reach the threshold. On the other side, if *thresh* is too small, the malicious nodes will be able to use the false positions and $t_{diff}$ values to cheat many legitimate users. In this part, we study the relationship between the selected threshold value and the node density in an attack-free environment. Its impacts on the detection accuracy will be investigated in the next subsection.
The simulation results are shown in Figure 3. In the X-axis we illustrate the average degree of connectivity of the secondary users in the network area. In the Y-axis we illustrate the percentage of the users that cannot generate at least *thresh* independent hyperbolas based on the information provided by the nodes under the same trustworthy sender. From the curves in Figure 3, we can see that a critical density exists for every threshold value. When the node density is larger than the critical value, the percentage of nodes that cannot reach the threshold will decrease very fast. This figure can provide very valuable information for us to determine the required node density for different threshold values when the proposed approach is deployed.

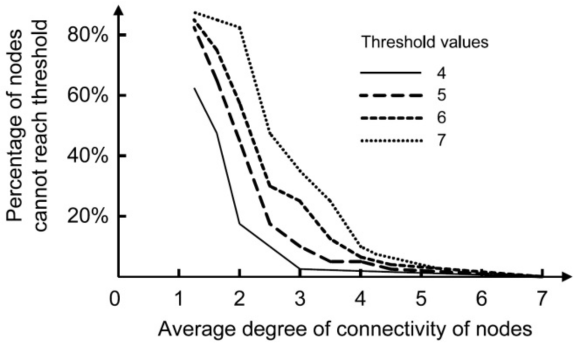

**Fig. 3. Selection of the threshold value.**

### *4.2.2. Detection accuracy of the proposed approach*

As we describe in Section 3.1, when the reference sender is trustworthy, the hyperbolas determined by the legitimate secondary users will pass through the real position of the interested sender. To mislead the legitimate secondary users, the attackers must provide false information about their positions and the $t_{diff}$ values. Below we investigate the impacts of the legitimate user density, the selected threshold value, and the attacker density on the detection accuracy. Since in Section 3.1 the analysis shows that the false positive and false negative alarms will follow the same curves, we illustrate only the false negative alarms in the figures.

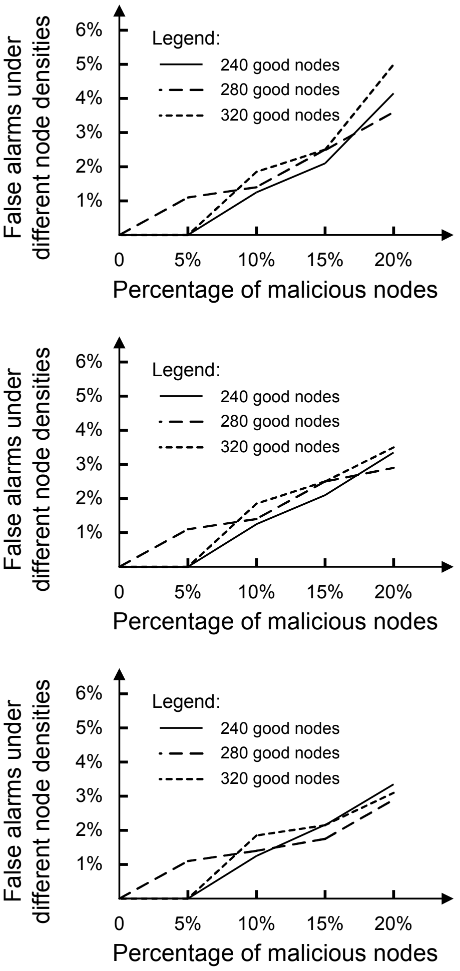

**Fig. 4. Detection accuracy under different threshold values and node densities. From top to bottom: the selected threshold values are 4, 5, and 6, respectively. For each curve, we have a constant number of legitimate users in the network and change the number of attackers.**

Figure 4 illustrates the false alarm rates under different node densities and threshold values. For each curve, we have a constant number of legitimate users (good nodes) in the network and we introduce different numbers of attackers. From the simulation results, we can see that for different node densities, their curves will stay close to each other when the percentage of attackers is the same. This can be explained as follows. As the density of the attackers increases, they can cheat more legitimate secondary users under the same threshold value. However, since the density of the legitimate users also increases, their ratio will stay the same. We can also find that when the threshold value increases, the false alarm rate starts to decrease since more attackers are needed to cheat a single legitimate user.

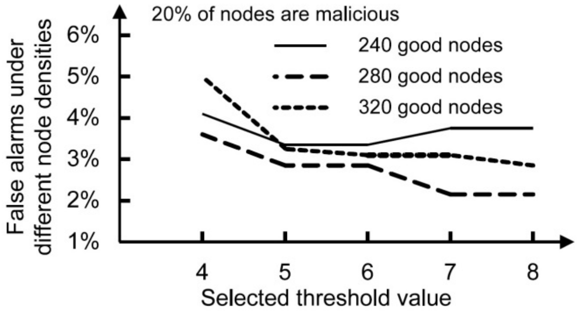

**Fig. 5. Detection accuracy under different threshold values.**

Figure 5 provides a better view of the relationship between the threshold values and the detection accuracy. Here we have a constant percentage of malicious nodes in the network. We can see that when there are 280 and 320 legitimate users in the network, the false alarm rate starts to decrease when the threshold value becomes larger. When there are 240 legitimate users in the network, the false alarm rate first decreases then increases again. This can be explained as follows. The average degree of connectivity under that scenario is 5.35. When the threshold value becomes too large (e.g. 8), many legitimate users will not be able to generate *thresh* hyperbolas passing through the real sender's position. The attackers can then use their fake hyperbolas to cheat more legitimate users.

## 5. Discussion

### 5.1. Why Depend on PNC to Measure Time Difference

As shown in Section 2, the proposed approach measures the starting point of interference of two sequences to estimate the distance between the nodes. Here we have to answer one question: why do not we directly use the GPS clocks or system clocks to measure the difference between the arrival time of the two sequences? In that way, we do not need the two sequences to interfere with each other and we can still allow the receivers to calculate the hyperbolas.

The following reason makes us use the physical layer network coding to measure the time difference. Previous research [25] has shown that wireless nodes have a maximum system clock drift rate at microsecond level ($10^{-6}$ second). At the same time, the deviations of clock drift rates are also at the microsecond level. Let us consider a wireless network that has the radio range $r$ = 250 meters. It will take the radio signal about $250m \div 300{,}000\ km\ per\ sec \approx 0.83 \times 10^{-6}$ second transmission time to reach the receivers. In this way, the measured duration and the clock drift are at the same level. Therefore, directly using the system clocks to measure the time difference will introduce a large error.

The GPS clocks are highly accurate (within several to tens of nanoseconds). However, when they are linked to a mobile device, the synchronization accuracy will become much worse because of the following reasons [19]. First, the GPS receiver can only supply timing information in discrete intervals (e.g. two times per second) to the device. This means the OS has to use its own timer for accessing real time information. Second, the accuracy of the system clocks and the processing capabilities of the devices will reduce the synchronization accuracy back to the microsecond level. For example, a GPS-based, stratum-1 level time server usually has an error at the microsecond level. This inaccuracy is good enough for the release of the hash chain entries for authentication but it is too loose for the measurement of signal interference.

### 5.2. Accurately Locate the Start Point of Collision

The detected starting point of signal interference could have affected the localization accuracy of the proposed approach. As shown in Equation (1), the wireless node depends on the starting point of collision to calculate $t_{diff}$. Considering the high propagation speed of the radio waves, if the detected collision is offset by several symbols, the introduced error can be large. To reduce the impacts of such errors, we can adopt the method described in [11]. Here each packet from the reference sender will start with a pilot bit sequence with known contents. Therefore, even when the detected collision has an offset of several symbols, we can still determine its correct starting point. Note that the pilot sequence has the length of 64 bits in [11] and it will not drastically increase the communication overhead.

## 6. Conclusion

In this paper we propose a PUE detection mechanism for cognitive radio networks based on physical layer network coding. The analysis shows that the difference between the starting points of interference at two receivers is restricted by the positions of the senders. Using a trustworthy node as the reference sender, we can determine multiple hyperbolas on which the interested sender resides. To turn this mechanism into a practical approach, we study several problems in the network. We design the PUE detection mechanism and study its overhead and the detection accuracy.

Immediate extensions to our approach consist of the following aspects. First, we will implement the proposed approach in software defined radio and test it in real network environments. Second, we will extend our approach to the environments in which the reference senders could be malicious. Finally, we will investigate using physical layer network coding to detect other attacks on wireless networks.

## Nomenclature

| | |
|---|---|
| $r$ | Communication range of wireless nodes |
| $d_{MN}$ | Distance b/w two nodes $M$ and $N$ |
| $T$ | A time moment |
| $t$ | A time duration |
| $s$ | Speed of radio signals |
| $t_{diff}$ | The difference between the arriving time of two interfered sequences |
| *thresh* | Threshold value to determine the sender's position |
| *Node identities* | |
| $TR$ | Trustworthy reference sender |
| $s_i (i = 1 \cdots p)$ | Legitimate secondary users |
| $m_j (j = 1 \cdots q)$ | Malicious attackers |
| $P$ | The real primary user |
| $U$ | The emulator |
| *Functions* | |
| $hash^j(y_i)$ | Hash chain of the random number $y_i$ |
| $Posi(m_j)$ | Real position of the attacker $m_j$ |
| $Posi(\overline{m_j})$ | Claimed fake position of the attacker $m_j$ |

## Acknowledgments

This research is supported in part by US NSF Award 1143602.

## References

[1] Federal Communications Commission. Facilitating opportunities for flexible, efficient, and reliable spectrum use employing spectrum agile radio technologies. ET Docket, (03-108), 2003.

[2] Pu D, Shi Y, Ilyashenko A, Wyglinski A. Detecting primary user emulation attack in cognitive radio networks. IEEE Global Telecommunications Conference (GLOBECOM), 2011, pp. 1–5.

[3] Yang T, Chen H, Xie L. Cooperative primary user emulation attack and defense in cognitive radio networks. International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), 2011, pp. 1–4.

[4] Liu Y, Ning P, Dai H. Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures. Proceedings of IEEE Symposium on Security and Privacy, 2010, pp. 286–301.

[5] Chandrashekar S, Lazos L. A primary user authentication system for mobile cognitive radio networks. Proceedings of the 3rd International Workshop on Cognitive Radio and Advanced Spectrum Management (COGART), 2010, pp. 1–6.

[6] Jin Z, Anand S, Subbalakshmi K. Mitigating primary user emulation attacks in dynamic spectrum access networks using hypothesis testing. SIGMOBILE Mobile Computing and Communication 2009; Rev. 13 (2):74–85.

[7] Leon O, Hernandez-Serrano J, Soriano M. Robust detection of primary user emulation attacks in IEEE 802.22 networks. Proceedings of the International Conference on Cognitive Radio and Advanced Spectrum Management, 2011, pp. 51:1–51:5. http://dx.doi.org/10.1145/2093256.2093307

[8] Chen R, Park J, Reed J. Defense against primary user emulation attacks in cognitive radio networks. IEEE JSAC, 26(1):25–37, 2008.

[9] Cabric D, Mishra S, Brodersen R. Implementation issues in spectrum sensing for cognitive radios. Proceedings of the Thirty-eight Asilomar Conference on Signals, Systems, and Computers, 2004.

[10] Jenkins R. Isaac. International Workshop on Fast Software Encryption, 1996, pp. 41–49. http://dx.doi.org/10.1007/3-540-60865-6_41

[11] Katti S, Gollakota S, Katabi D. Embracing wireless interference: analog network coding. ACM SigComm, 2007, pp. 397–408.

[12] Stevenson C, Chouinard G, Lei Z, Hu W, Shellhammer S, Caldwell W. IEEE 802.22: the first cognitive radio wireless regional area network standard. ACM Communication Magazine, 2009; 47 (1): 130–138. http://dx.doi.org/10.1109/MCOM.2009.4752688

[13] IEEE 802.16 License-Exempt (LE) Task Group, IEEE 802.16 Draft Version 15, 2010.

[14] Wang W, Pu D, Wyglinski A. Detecting sybil nodes in wireless networks with physical layer network coding. IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2010, pp. 21–30.

[15] Li Z, Pu D, Wang W, Wyglinski A. Node localization in wireless networks through physical layer network coding. Proceedings of IEEE Global Communications Conference (GLOBECOM), 2010, pp. 1–5.

[16] Perrig A, Canetti R, Tygar J, Song D. The tesla broadcast authentication protocol. RSA Crypto Bytes 2002; 5 (2) : 2–13.

[17] Perrig A, Tygar J, Song D, Canetti R. Efficient authentication and signing of multicast streams over lossy channels. Proceedings of IEEE Symposium on Security and Privacy, 2000.

[18] Song D, Tygar J, Zuckerman D. Expander graphs for digital stream authentication and robust overlay networks. Proceedings of IEEE Symposium on Security and Privacy, 2002.

[19] Sterzbach B. GPS-based clock synchronization in a mobile, distributed real-time system. Real-Time Systems, 1997, 12(1):63–75. http://dx.doi.org/10.1023/A:1007910115824

[20] Tippenhauer N, Popper C, Rasmussen K, Capkun S. On the requirements for successful GPS spoofing attacks. Proceedings of the ACM conference on Computer and communications security (CCS), 2011, pp. 75–86.

[21] Leonardi M, Mathias A, Galati G. Two efficient localization algorithms for multilateration. International Journal of Microwave and Wireless Technologies, 2009; 1: 223–229.

http://dx.doi.org/10.1017/S1759078709000245

[22] Wu H, Lu I. A simple and accurate linear solver for hyperbolic localization. IEEE wireless communications and networking conference, 2005.

[23] Doukhnitch E, Salamah M. General approach to simple algorithms for 2-D positioning techniques in cellular networks. Computer Communications, 2008; 31 (10): 2185–2194. http://dx.doi.org/10.1016/j.comcom.2008.02.005

[24] Mitsa T, Parker K. Digital halftoning using a blue-noise mask. Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 1991.

[25] Song H, Zhu S, Cao G. Attack-resilient time synchronization for wireless sensor networks. Proc. of IEEE MASS, 2005, pp. 765–772.