JENNA: a Jamming Evasive Network-coding Neighbor-discovery Algorithm for Cognitive Radio Networks

Alfred Asterjadhi‡ and Michele Zorzi‡
‡Department of Information Engineering, University of Padova
via G. Gradenigo 6/B, I-35131, Padova (PD), Italy
E-mail:{aasterja, zorzi}@dei.unipd.it

Abstract—In this paper we address the problem of neighbor discovery in cognitive radio networks. Cognitive radios operate in a particularly challenging wireless environment. In such an environment, besides the strict requirements imposed by the opportunistic co-existence with licensed users, cognitive radios may have to deal with other concurrent (either malicious or selfish) cognitive radios which aim at gaining access to most of the available spectrum resources with no regards to fairness or other behavioral etiquettes. By taking advantage of their highly flexible radio devices, they are able to mimic licensed users behavior or simply to jam a given channel with high power. This way these concurrent users (jammers) are capable of interrupting or delaying the neighbor discovery process initiated by a normal cognitive radio network which is interested in using a portion of the available spectrum for its own data communications.

To solve this problem we propose a Jamming Evasive Network-coding Neighbor-discovery Algorithm (JENNA) which assures complete neighbor discovery for a cognitive radio network in a distributed and asynchronous way. We compare the proposed algorithm with baseline schemes that represent existing solutions, and validate its feasibility in a single hop cognitive radio network.

I. INTRODUCTION

Cognitive Radios (CRs) [1] are a promising technology likely to be deployed in the very near future to alleviate the spectrum shortage problem faced by traditional wireless systems [2], [3]. They are equipped with highly flexible RF front-ends, which can dynamically change their transmission parameters in order to optimally exploit the available spectrum resources at a given geographical area and time.

This increased capability of CRs to self adapt makes them the perfect candidates for opportunistic spectrum access in those bands which are assigned to licensed users (primary radios, PRs) on a long term basis. While these PRs are enabled to access the licensed spectrum resources anytime and anywhere, within the contractual limits imposed by spectrum management authorities, CRs have to be very cautious while accessing these resources. That is, CRs have to scan and identify unused spectrum resources in the licensed bands and most importantly they have to guarantee not to interfere with PRs that are operating in their legitimate spectrum resources.

It is well known that this scenario per se is very challenging while deploying a Cognitive Radio Network (CRN) which is going to coexist with one or several PRs in the area [4]. However, when considering the natural evolution of cognitive radio networks to more complex systems, the challenges and problems to be faced increase dramatically [5], [6]. More specifically, the inherent capability of CRs to base their decisions on their “view” of the environment and to learn from experience makes their operation susceptible to a variety of malicious attacks. External or internal modification of their wireless environment perception may result in sub-optimal (or even denial of) operation in a given licensed spectrum. This may happen when multiple CRs are accessing the same limited spectrum resources and any of them, in order to satisfy its bandwidth demands for data communication, can act improperly to gain exclusive access to the available resources. Techniques such as Primary user emulation attacks [7] can be adopted by such users to achieve their goal. In addition, there can be cases where malicious users are interested in simply interrupting other CRNs’ operations by interfering with their communications. The transmission of jamming signals in a given spectrum band is the most effective technique to disrupt a communication which has been or is going to be established in that spectrum band [8].

In this paper we focus on the first step that a CRN has to undertake in order to operate, i.e., the neighbor discovery phase. Every CR independently scans the available spectrum resources and maintains a list of channels which are available for communications. Moreover, it gathers additional information for the correct utilization of the spectrum such as channel occupancy, primary users identification, selfish CRs that might mimic licensed users behavior [7], etc. Once the CRs have this information, any of them can wake up and begin sending control packets in order to discover and disseminate the acquired information to its neighbors. We use network coding [9] to disseminate these control packets in an efficient and reliable way, making it possible to have substantial gains in terms of dissemination delay and robustness with respect to existing schemes. The combination of network coding with random channel hopping makes it possible to obtain an effective neighbor discovery algorithm that enables the deployment of CRNs in challenging wireless environments in a totally distributed and asynchronous way. Hence the designed protocol fits very well in the next generation wireless networks paradigm, where cognitive devices should adapt in the best possible way to the wireless environment conditions.

The paper is organized as follows. In Section II we present the related work on neighbor discovery algorithms for cognitive radio networks. In Section III we describe the network model discussing the structure and capabilities of both normal CRs and jammer CRs. Section IV introduces the proposed system architecture and a detailed description of its main components, concluding with a representative
example of the algorithm execution. In Section V we present some performance evaluation results for the protocol with respect to baseline schemes that represent the behavior of existing neighbor discovery protocols. The paper concludes in Section VI with a discussion of the benefits obtained by our algorithm and possible future work.

II. RELATED WORK

In order to deploy a CRN, nodes have to discover and exchange information such as neighborhood and spectrum availability with their neighbors. The neighbor discovery process starts when a node wakes up and begins to broadcast beacons and ends when it receives replies from all its neighbors that are within transmission range. In traditional ad hoc networks this process is easily implemented as all nodes are tuned on the same channel, which makes it possible to rapidly exchange all the required information with all neighbors [10]–[14].

Unfortunately, in cognitive radio networks the neighbor discovery process becomes more challenging because CRs operate over a set of multiple channels which may vary from node to node, depending on their proximity to PRs. To tackle this problem there have been several proposals in the literature which are based on deterministic [15]–[17] or randomized algorithms [18].

In particular, in [15], [16] the authors propose neighbor discovery algorithms for time-synchronous networks which assure neighbor discovery in $O(MN)$ and $O(Mn \log(N))$ time slots, respectively, where $M$ is the maximum number of channels, $n$ the number of nodes and $N$ the dimension of the label space from where nodes obtain their ids. Another solution is presented in [17] where the algorithm does not require nodes to be globally time-synchronized among them. In this proposal the time required to elect a leader which subsequently discovers all neighbors is $O(NM^2)$. We note that these solutions, being based on deterministic algorithms, are very susceptible to jamming attacks, as it is very easy to disrupt neighbor discovery once the jammers know the channel hopping pattern followed by the CRs, which is a simple round-robin in these proposed solutions. Moreover, the fact that neighbor discovery latency depends on the dimension of the label space $N$ makes these solutions not appropriate for networks with a small number of nodes and high $N$.

A different approach is proposed in [18]. Here the authors assume that nodes are globally time-synchronized (e.g., nodes equipped with GPS modules). They consider different frequency hopping patterns from single frequency hop to random hop patterns without giving neighbor discovery termination guarantees. We note that the best strategy to follow when the CRN suffers jamming attacks is to randomly hop across all channels. In Section IV this approach with the random hopping pattern will be represented by the SLF scheme where we assume CRs know the number of nodes, $n$, in order to determine when to conclude the neighbor discovery process.

As to neighbor discovery in the presence of jammers, there has been some research in the case of traditional wireless networks [14], [19], [20] while for multi-channel cognitive radio networks, to the best of the authors’ knowledge, there are no proposed solutions to this problem so far. Hence, JENNA is the first solution to the neighbor discovery problem in CRN which takes into account the presence of jammers. The proposed algorithm has the following benefits: 1) it is fully distributed, 2) it does not need global time-synchronization among nodes, 3) it assures fast and simultaneous neighbor discovery for all nodes with high probability, 4) its dissemination performance does not depend on the label space dimension $N$, but rather on the actual number of nodes $n$ in the network, and 5) it is very robust to jamming attacks.

III. NETWORK MODEL

In this section we introduce the network model which will be used throughout the rest of this paper. The electromagnetic spectrum is given under license to $P$ primary users which can access it at any given time and frequency. This spectrum band is divided into $M$ non-overlapping orthogonal channels $C_{tot} = \{1, \ldots, M\}$ available for both primary and CR communications.

A. Normal cognitive radio model

CRs can access any of the licensed channels $c \in C_{tot}$ every time they can assure not to interfere with any of the PRs activity. To achieve this, they are capable of sensing the available spectrum with techniques that will be mentioned briefly in Section IV-B. CRs are assigned a unique identifier $i \in \{1, \ldots, N\}$ and are equipped with a single transceiver. Hence they can either transmit or receive on a single channel at any given time.

We further assume a time-slotted system where CRs do not require global time synchronization among them. They only require to have similar clock ticks in order to be able to synchronize at the slot boundaries [21].

B. Jammer cognitive radio model

Adversary nodes can launch different types of jamming attacks [5], [6]. Their goal is to deny channel availability for the longest period of time, potentially delaying or totally disrupting the neighbor discovery process initiated by a CRN. For simplicity we divide adversaries into two categories:

Static jammers continuously emit radio signals in a given frequency for a long period of time. The most representative example is the Primary User Emulation attack (PUE) launched by selfish CRs which want to reserve the spectrum for their own communications. By mimicking PRs’ signal characteristics these CRs mislead normal CRs into concluding that the spectrum is occupied by legitimate PRs. It is important for normal CRs to gather as much information as possible to detect such jammers and react consequently [7]. In this category also fall most of the attacks described in the context of single channel wireless networks [8].

Reactive jammers jam a single channel at any time following a random frequency hopping pattern trying to disrupt normal CRs communications in the available spectrum $C_{tot}$. In this category fall malicious CRs [22] that exploit their software defined radios to transmit high power spectral density signals in a random channel $c \in C_{tot}$, disrupting any ongoing transmission. While their signal properties make reactive jammers easy to detect, their random channel hopping pattern makes
them very difficult to deal with. This is because normal CRs cannot extract any information based on previous observations of their jamming activity. Hence they cannot adopt intelligent anti-jamming techniques to avoid their attacks.

Let $j^*_i \in J^s$ represent a static jammer transmitting on channel $c$ and $j^r_i \in J^c$ a reactive jammer transmitting over a random hopping sequence with hopping frequency comparable to the slot duration.

For simplicity, we assume that all jammers have the same communication range $R_j$ and that any CRs which happen to be within this range, tuned at the jammed frequency in a given slot, will receive unrecoverable packets, i.e., all packets interfered by a jammer will be considered to be lost.

### IV. JENNA: System architecture

In this section we present the system architecture of the proposed jamming evasive network-coding neighbor-discovery algorithm. We first give a generic description, focus on its main components and then give an example of the algorithm execution in a simple CRN.

#### A. Generic description

CRs that want to perform neighbor discovery wake up, enter the passive mode, and start the spectrum scanning phase. They switch over all the channels in $C^{\text{tot}}$, collecting information regarding PRs' activity, static jammers and other information that might be useful during network setup.

When a CR decides to start neighbor discovery, it enters the active mode, synchronizes in a random channel, and schedules its control packet for transmission. The CR transmits in a given slot if it senses the channel free, otherwise it defers the transmission. This first packet triggers the transition phase. During this phase CRs are divided into two groups, those in active and those in passive mode. CRs in passive mode switch to active whenever they receive a control packet from a generic neighbor CR. Active CRs hop randomly over the free channels, and schedule for transmission linear combinations of the control packets they have received and stored in their buffers so far. Every time a CR receives a control packet it performs Gaussian elimination on the decoding matrix. If the matrix has full rank the CR can decode the packets in its buffer, hence it can activate a timeout period during which it continues transmitting packets to help neighbor CRs that may not have been able to decode their packets yet. This way the algorithm guarantees that nodes are able to decode all the control packets that are stored in their buffers with high probability [23].

Depending on the way CRs use the timeout value $T_{\text{out}}$ during the termination phase, we implement two different versions of the algorithm, namely, asynchronous and synchronous. In the first case, CRs use the estimated timeout period to disseminate packets to neighbors that have received only partial information so far. In the second case, CRs synchronize to the same timeout value making it possible to end the neighbor discovery process at the same time slot for all CRs.

#### B. Spectrum scanning phase

We assume that initially all CRs in the network are in passive mode. After a CR wakes up it stays in the scanning phase until it decides or is requested to perform neighbor discovery by some nearby node. During this phase, CRs scan the set of all the available channels $C^{\text{tot}}$, following a random hopping pattern, detecting independently the existence of PRs activity and of possible primary user elimination attackers (more generally static jammers)\(^1\). This can be achieved by using spectrum sensing techniques, such as energy detection, cyclo-stationary feature detection or matched filter detection [24]. The problem of detecting PUE attackers can be tackled with techniques such as [25]. At the end of this phase, each node $i$ in the CRN has created a list of free channels $C_{i}^{\text{free}} = C^{\text{tot}} \backslash (P_i \cup J^s_i) = \{c_1, c_2, \ldots, c_{K_i}\}$ that $i$ can use for communications. This list will be included in the control packet along with additional information such as which channels are used by legitimate PRs, PUE attackers, etc.

#### C. Transition phase

This phase starts when a generic CR $i$ begins neighbor discovery sending its control packet randomly in a channel $c \in C_{i}^{\text{free}}$. Every time CRs transmit or receive control packets they enter the active mode. The channel hopping pattern used by the algorithm is a random sequence with a generation seed that is calculated in real time. This way reactive jammers cannot disrupt CRs' communications even if they gain access to the internal memory of a CR where predefined generation seeds are stored. We note that, during the transition phase, there is dynamic diversity among CRs. A portion of CRs are in passive mode where they are still sensing the available channels, and the rest of them are in active mode disseminating the control packets. This phase is very delicate as it is the moment when reactive jammers are likely to get activated to counteract the neighbor discovery process initiated by active CRs. This is because nodes in passive mode may find their sensing data misleading as they will sense activity caused by reactive jammers in terms of short impulses, hence probably including these channels as not free for communication. However, considering the different characteristics of static and reactive jammers, a node may still be able to distinguish among them, making a more selective channel occupancy categorization.

The transition phase ends when CRs are all in active mode. For the estimation of $T_{\text{out}}$ and $n$, CRs keep updating their local variables based on the information they receive along with the encoded packets. This way, prior to the termination phase, CRs will have a $T_{\text{out}}$ value which is equal to the time required to disseminate the transition phase duration to all CRs.

\(^1\)In our framework we assume that reactive jammers are not active during this phase. This is to reflect the fact that reactive jammers become active exclusively based on an external excitement i.e., a generic CR begins to broadcast packets for neighbor discovery [8]. Once the jammer hears activity in that frequency, it initiates jamming the spectrum to disrupt the ongoing communications that are being established by normal CRs.
D. Packet transmission schemes

Let \( n \leq N \) be the number of CRs which are going to participate in the neighbor discovery process, where \( N \) is the dimension of the label space for assigning ids to the CRs. Let \( i \in \{1, \ldots, n\} \) denote a generic CR and \( x_i \) the control packet that it has generated at the end of the spectrum sensing phase. We have that in every slot \( s \in \{1, \ldots, S\} \) the transmission scheme can be modeled as:

\[
\begin{bmatrix}
    y_1 \\
    \vdots \\
    y_n
\end{bmatrix} =
\begin{bmatrix}
    g_{1,1} & \cdots & g_{1,n} \\
    \vdots & \ddots & \vdots \\
    g_{n,1} & \cdots & g_{n,n}
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    \vdots \\
    x_n
\end{bmatrix} = G
\begin{bmatrix}
    x_1 \\
    \vdots \\
    x_n
\end{bmatrix}
\]

(1)

where \( G \) is the system transfer matrix in slot \( s \), and \( y_i \) is the control packet which can be transmitted during slot \( s \) by CR \( i \). As previously discussed, we will consider the following transmission strategies:

- Random Message Selection (RMS): each CR transmits a packet randomly selected among those received so far. In our model, this is represented with a random permutation matrix \( G = P \) which is a square matrix with all zeros and a 1 value per row in a random column.
- Self Replication Strategy (SLF): each CR transmits only its own control packet. In this case \( G \) is the identity matrix \( I_n \).
- Network Coding (\( GF(2^q) \)): each CR transmits a random linear combination of the packets received so far, including its own. The matrix contains random coefficients over the Galois field \( GF(2^q) \). The packet format is shown in Fig. 1.

E. Termination phase

For all transmission schemes we have that the dissemination ends when all CRs receive all the packets generated in the CRN. For the algorithmic implementation of the RMS and SLF schemes, we assume that CRs know in advance the number \( n \) of CRs in order to end the neighbor discovery process.

In the case of our protocol, this is not necessary as the termination condition is based on the joint estimated \( n \) value during the transition phase and the rank of the decoding matrix. Hence, CRs can terminate neighbor discovery only if they have full rank i.e., can decode all the received packets. As mentioned previously, we implemented two versions of the algorithm which have different termination conditions. When using the asynchronous algorithm, CRs consider the discovery process terminated once they can decode all the packets in their buffers. However they continue to transmit additional packets for \( T_{out} \) to help other CRs that are still unable to decode their packets. The same thing happens for the synchronous case with the only difference that the \( T_{out} \) is shared among all nodes, including it in the packets header, and its value is dictated by the last CR that has been able to decode all the received packets.

F. Description by example

To briefly describe the scheme, in Fig. 2 we show the execution of the asynchronous version of the JENNA protocol in a simple network consisting of 10 CRs which have to coexist with one PR and a reactive jammer active in the area. With reference to Fig. 2, at the beginning, all CRs are in passive mode, scanning the available channels to detect the presence of PRs or static jammers in the area. When the sensing phase finishes, the nodes have their channel list \( C_{i,free} = \{1, 2, 3, 5\} \) for \( i \in \{1, \ldots, 10\} \). Note that in general the list of available channels may be different for each node. At a given moment, slot \( s = 1 \), node 8 awakes and becomes active, sending its control packet on channel 1. The transmission of the packet by node 8 wakes up \( j^1_{\text{reactive}} \), which happened to be sensing in channel 1. Hence, \( j^1_{\text{reactive}} \) initiates random jamming over all channels with jamming duration equal to the slot length. In slot 2 the transmission of node 8 successfully reaches nodes 5 and 10, which then enter the active state. In slot 3 we have node 5 transmitting in channel 5. The control packet generated by node 5 contains a linear combination of its control packet and the one it received in the previous slot from node 8. Following this procedure, in slot 10 all nodes are active and disseminating information to each other, hopping randomly in different channels and avoiding to transmit their control packet every time they happen to fall in the jammed frequency. The neighbor discovery process ends when all nodes in the CRN are able to decode the information they have stored in their buffers so far, i.e., the decoding matrix has full rank.

V. Performance evaluation

In this section we evaluate and compare the proposed neighbor discovery algorithm (synchronous and asynchronous), with respect to the above mentioned baseline schemes, SLF and RMS. We simulate a CRN with different numbers of CRs involved in the neighbor discovery process sharing a set of \( M = 30 \) channels with primary users, static and reactive jammers.

Nodes are all within transmission range of each other and the set of available channels sensed by CRs is the same.
Hence after the sensing phase CRs hop over a similar set of available channels. This is because PRs are assumed to have a transmission range which is higher than that of CRs. Moreover the spectrum is subject to frequent attacks by reactive jammers which, once they have detected CR activity on a channel, begin jamming randomly the available channels to disturb the neighbor discovery process. We assume that reactive jammers, after being activated, do not sense the medium before transmitting their jamming impulse, hence on a given frequency there can be more than one jammer in a given slot.

A. The impact of coding on dissemination latency

We hereby describe the impact of coding on the dissemination delay performance. This is shown in Fig. 3 where we plot the number of slots required for all nodes in the network to discover their neighborhood, i.e., dissemination delay, against different neighbor discovery schemes. As we can see, using network coding provides faster dissemination of the control packets, making it possible to finish the neighbor discovery in less time. The achievable improvement with respect to SLF ranges from 3 to 6 times in these settings. Regarding the coding performance we note that coding over Galois fields of higher size does not bring significant benefits in terms of dissemination delay, except for the case of a small number of CRs, where packet diversity is highly required. As an example, for \( n = 10 \), the gain obtained when using coding over GF(256) instead of GF(2) is around 20%. The lack of further gains from higher coding sizes comes from the fact that the random hopping pattern that CRs follow is already able to provide most of the required diversity for a fast neighbor discovery process.

B. The impact of free channels and number of CRs

In Figs. 4 and 5 we show the dissemination delay as a function of the number of free channels and the number of CRs in the network, respectively. More specifically, in Fig. 4 it can be noted how the dissemination delay increases for all the schemes as the number of free channels increases. This is because with a wide range of free channels it is more likely that a node tunes to a channel where there are no other nodes for control packets exchange. Note that, while for RMS and SLF schemes increasing \( C_{\text{free}} \) leads to a longer dissemination delay, in the case of network coding there is a local minimum which assures lowest dissemination delay for a given \( n \). Hence, given the number of nodes \( n \), there exists an optimal number of channels \( C_{\text{free}} \) that nodes can access in order to maximize the rate with which their buffers rank increase, leading to a minimum dissemination delay.

In Fig. 5 we can see the same behavior in terms of dissemination delay with respect to the number of nodes which are involved in the neighbor discovery process. However, we note that network coding is particularly robust in terms of dissemination delay for varying number of CRs in the network, providing comparable performance for a wide range of number of nodes, which is not true in the case of RMS and SLF that suffer particularly in those scenarios where the neighbor
discovery has to be performed over a high number of nodes. This is because, with a high number of nodes in the network, the number of nodes per channel increases and hence packets are disseminated to more nodes in each slot. This in turn, when using network coding, increases the diversity in packet mixing, leading to faster control packet dissemination.

We also note that in all cases the synchronous version of the neighbor discovery algorithm takes more time to terminate. This is because nodes need not only to decode the packets but also synchronize to the same value of $T_{out}$.

C. The impact of reactive jamming attacks

To conclude, we show the performance of the algorithm with respect to the number of reactive jammers which are accessing the available channels randomly over frequency and time. In all cases network coding dissemination is faster with respect to RMS and SLF, with gains that are 6 and 4 times, respectively. As expected the synchronous version requires additional time to provide simultaneous termination for all nodes.

![Figure 6](image.png)

Fig. 6. Dissemination latency versus number of simultaneous reactive jammers $J^*$ for $n = 20$.

VI. CONCLUSIONS

In this paper we present a novel neighbor discovery algorithm which exploits network coding for fast and reliable control packet dissemination for a successful discovery of other nodes in the area. We compare it with baseline schemes representing state of the art solutions, and provide simulation results to show the benefits of our approach.

Future work is focused on more intelligent channel hopping sequences, implementation in real devices and the development of an extended neighbor discovery algorithm which aims at joint neighbor discovery and cluster formation for a cognitive radio ad hoc network.

ACKNOWLEDGEMENT

This work was partially supported by the European Commission through the ARAGORN project (FP7 ICT-216856).

REFERENCES


