PHY/MAC Signalling Protocols for Resilient Cognitive Radio Networks

Martin Peres
LaBRI, University of Bordeaux
Talence, France
Email: martin.peres@labri.fr

Mohamed Aymen Chalouf
IRISA, University of Rennes 1
Lannion, France
Email: mohamed-aymen.chalouf@irisa.fr

Francine Krief
LaBRI, University of Bordeaux
Talence, France
Email: francine.krief@labri.fr

***Abstract*—Our society relies more and more on wireless communication technologies while most of the RF spectrum has already been allocated by the states. As a result, un-licensed bands are becoming crowded which makes it difficult to create a reliable network without using more spectrum than really necessary. Allowing radio nodes to seamlessly switch between different frequency bands without prior synchronisation would allow the creation of a truly resilient radio network capable of avoiding the frequency bands used by nodes that are not part of the network. In this paper, we propose using software-defined radios in order to sense the surrounding RF environment to find the most suitable bands for communication. We also propose a PHY-layer and a MAC-layer signalling protocols to provide a seamless way of discovering other nodes and selecting the parameters that will be used for communicating with them. Our first experimentation results are very promising towards defining a resilient cognitive radio network.**

I. Introduction

Today's society is increasingly reliant on wireless technologies for human and machine-to-machine communications. Due to their lower cost, ease of deployment and the allowed mobility, we can expect this trend to continue.

The radio frequency (RF) spectrum used for communications can be shared by multiple radio transceivers by transmitting at different frequencies (FDMA/OFDMA), at different times (TDMA) or using different codes (CDMA). The frequency at which a transceiver is allowed to radiate (transmit power) is currently state-regulated and licenses are given or sold for a technology, set of regions and a company. So-called unlicensed bands, where the state allows individuals to radiate without authorisations, come with limitations such as the maximum emission power, duty cycle and bandwidth in order to provide a good probability of success for everyone.

Evading crowded or perturbed frequency bands is difficult on a traditional radio because it is not possible to be both available to receive messages and to look for non-crowded frequencies (sensing). This is however possible using software (SW) radios because they do not demodulate the signal themselves. They merely sample the voltage found at the antenna which allows computers connected to them to listen to any frequency band available in a much larger tunable frequency range. For instance, Nuand's bladeRF, can receive or emit signals in a frequency band of up to 28MHz with a central frequency ranging from 300MHz to 3.8GHz. A computer can then perform the sensing operation to detect and decode multiple transmissions happening simultaneously in this 28MHz window. Nodes analysing their environment are called cognitive (radio) nodes (CR) and form a Cognitive Radio Network (CRN). CRs may re-use licensed bands if they do not interfere with the licensed users (primary users).

In this paper, we propose a PHY- and a MAC-layer signalling protocol to take advantage of the capabilities of SW radios in order to create a communication channel resilient to unintentional jamming. Contrarily to the current state of the art, nodes using our protocols can perform sensing while still being available to other nodes, allowing new nodes to join the cognitive network at any time. Section II introduces the state of the art relative to cognitive nodes. Detecting transmissions using SW radios is detailed in section III. Our propositions for the PHY and MAC signalling protocols are respectively found in sections IV and V. We conclude in section VI and present the future works.

II. State of the art

The simplest form of cognitive radio network uses a common control channel (CCC) that is used to initiate communications and exchanging sensing information [1][2]. The drawback of using a CCC is that it presents a single point of failure and does not scale well with multi-hop ad-hoc networks. It is thus better not to use a CCC although it presents other difficulties [3]. Indeed, in the absence of a CCC, transmissions could happen anywhere in the RF spectrum. In order to communicate, two transceivers first need to find each others by using a "blind rendez-vous" technique [4] which can guarantee two receivers will find each others if they have one available channel in common. However, nodes willing to rendez-vous should all follow the jump sequence given by the algorithms. This is not practical in a CRN because nodes would loose the ability to communicate while sensing for new nodes.

In [5], the proposed MAC protocol provides a decentralised and CCC-free rendez-vous mechanism which also uses the Signal-to-Noise (SNR) ratio in order to use the fastest modulation achievable. This approach is the most resilient one we found but it assumes that nodes already know the list of the surrounding nodes and that a channel list is shared between them. The first assumption can be satisfied by adding a beaconing mechanism where CRs broadcast their presence. However, the second assumption is problematic because SW radios usually require knowing the central frequency of a transmission to be able to decode it. Not having a channel list available means a different way of detecting transmission needs to be used.

Implementing MAC protocols in a SW radio has additional challenges that come from the fact that samples are not processed close to the ADC like in usual radios [6]. Samples are usually sent to a computer using a non-real-time communication medium, such as USB or Ethernet, and processed on a non-real-time operating system. A lot of buffering is thus necessary to keep the radio fed with the samples. This buffering increases the latency and the variability of the emission and reception time. Signalling protocols in CRNs need to be able to cope with this variability.

III. Detecting transmissions in Software radios

The way transmissions can be detected and decoded using SW radios is quite different from traditional radios. Since our propositions heavily make use of these differences, we briefly introduce spectrum sensing and our software infrastructure.

A. Time Domain vs Frequency Domain

Traditional radios decode one transmission at a time. They tune to a central frequency, set the expected PHY parameters and wait for the received power to be higher than a threshold that depends on the noise of the radio.

With SW radios, it is possible to decode multiple transmissions happening at the same time provided we can isolate them frequentially. By using the Fast Fourier Transform (FFT), we can convert the radio's samples to the frequency domain. The output of an FFT is the received power and phase at various frequencies. The frequency resolution is linear with the number of samples used to compute the FFT.

Although thermal noise is often considered Additive, White and Gaussian (AWGN), we experimentally found out that when using Ettus Research's USRPs SW radios, the noise distribution in the frequency domain mapped perfectly to an extreme value distribution, given in Eq. 1, with $\mu = 1.67$ and $\sigma = 4.5578$. However, the average of the distribution depends on the central frequency and the offset to it in the spectrum window. By using this model, detecting transmission only requires learning the noise's average power at every frequency bin in order to find a threshold over which it means additional power has been received. The end of the transmission is detected when the bin's power stays under its threshold for more than n (Eq. 3) samples which depends on the confidence level wanted that a transmission x dB over the noise will be correctly detected. Detection probability in a low-SNR scenario could be increased using a cyclo-stationary analysis [7].

$$f(x|\mu,\sigma) = \frac{1}{\sigma} e^{\frac{x-\mu}{\sigma}} e^{-e^{\frac{x-\mu}{\sigma}}} \quad (1)$$

$$p(x|\mu,\sigma) = 1 - e^{-e^{\frac{x-\mu}{\sigma}}} \quad (2)$$

$$p(x|\mu,\sigma)^n > p_{confidence} \Leftrightarrow n = \frac{ln(1-p_{confidence})}{ln(p(x|\mu,\sigma))} \quad (3)$$

B. Decoding and collecting statistics

We are now able to detect the beginning and the end of transmissions in the frequency domain. We propose to use it to fill a table called the Radio Events Table that contains PHY-layer metadata about transmissions such as the start/end time, frequency band and the received power at the antenna. Using this information, a decoder reads the samples that contain the transmission, apply a pass-band filter to only let this transmission pass, roughly correct for the massive frequency offset and feed this to the usual demodulators used in software-defined radios. The demodulated frame can then be used to add additional information to the Radio Event Table such as the source node's ID and its nature (primary or secondary user). Figure 1 presents an overview of the sensing/receiving process.

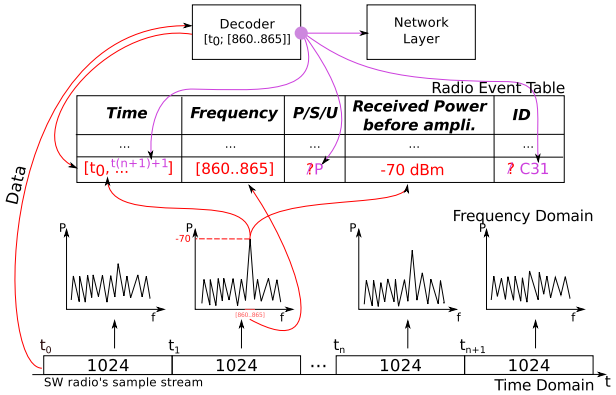

Figure 1: Overview of the sensing process - Filling the Radio Event Table from the software radio's sample stream

These experimentations have been used as part of the LICoRNe project, funded by the French National Research Agency (ANR), in order to create a video stream between two CRs that switch channels when a primary user appears in the current channel. Additional experimentation on transmission detection, demodulation and transmissions in the time domain have also been used by 8 students for building a smart box capable of detecting its surrounding wireless sensors and automatically finding their modulations. Our experimentations are available at https://github.com/mupuf/hachoir.

IV. PHY-layer signalling protocol

CRNs require information about the network topology, available channels, channel fading and on which frequency bands any surrounding CR is available in order to provide efficient unicast or broadcast schemes. Unfortunately, the schemes proposed in the state of the art try to reduce the requirements to provide connectivity at the expense of spectrum utilisation, delay and throughput. Moreover, blind rendez-vous, unicast MAC protocols and broadcast protocols are not compatible as they all require to be responsible for selecting the frequency band the radio will listen and send on.

The goal of our PHY-layer signalling protocol is to allow CRs to find each others (rendez-vous) and to know when and how to contact each others in the future, even if CRs are hopping from frequency bands to frequency bands. To do so, CRs should have at least part of their frequency hopping pattern predicable. Since CRs can be available on a limited amount of bands without drastically increasing the maximum latency to contact them, advertising the actual frequency hopping pattern explicitly is possible without creating a very large beacon frame. Once the hopping pattern and the current position in the pattern of a CR is sent in a frame, nodes that received it can predict when and on which band they

should send transmissions to this CR. To make this time synchronisation more accurate, the emitting and receiving CRs should compensate for the delay introduced by the kernel and the radio in the propagation of the emitted or received samples. In a situation where no under-runs happen and the radio is emitting/receiving a constant stream of sample, the TX delay is entirely predicable due to the fixed sampling rate. However, the average RX delay needs to be specified. An example beacon is given below in its ASCII form.

```
<beacon_frame>{ node_id=23, tx_pwr=10dBm,
[
  { {band1}, len=0.4, period_offset=0.0 },
  { {band2}, len=0.4, period_offset=0.0 },
  { {band3}, len=0.3, period_offset=0.5 },
],
period=1000ms, cur_period_offset=0.126 }
```

In this example, the beacon is sent by node_id 23 which is available on 3 bands. At the beginning of the hopping cycle, node 23 is available on both $band1$ and $band2$ for 400ms ($0.4 * 1000ms$). At 500ms ($0.5 * 1000ms$), node 23 will be available on $band3$ for 300ms. Node 23 is currently 126ms ($0.126 * 1000ms$) in its hopping pattern cycle which means the radio will stop being available on bands 1 and 2 in 274ms (400ms - 126ms). One slot of 100ms (400ms to 500ms) and one of 200ms (800ms to 1000ms) are not currently allocated in the beacon. This allows the radio to either put itself to sleep or perform sensing anywhere in the tunable frequency spectrum. This beacon has been sent at 10 dBm, this allows the receiver to compare it with the received power to evaluate the channel fading so as CRs can lower their transmission power if they assume the fading is roughly symmetric.

A. Bootstrapping

We propose that when booting up, CRs should first sense the tunable spectrum in order to find frequency bands that are usable without disrupting primary users. During this sequence, the transceiver is passive and should linearly scan the tunable spectrum. If beacons are received from cognitive users, they should be stored in a neighbours CR list.

B. Advertising

Once some frequency bands have been found to be available, we propose a CR should send its beacon when:

1) It becomes available on a frequency band;
2) No beacon has been sent on this band for some time;
3) Another CR requests the CR to resend it;
4) Another CR requests every CR to resend it.

Rules 1) and 2) generate a background traffic that allows CRs to keep their hopping patterns synchronised without needing an external time-synchronisation mechanism. Rule 3) allows CRs to re-synchronise with one another and to check if a CR has left from the band. Rule 4) enables a CR to request all his surrounding CRs to get the list of currently available CRs on the current band to ease the discovery process.

C. Scanning for other cognitive nodes

Our PHY-layer proposition is meant to make scanning as simple and un-intrusive as possible for CRs. Information about surrounding CRs is gathered simply by receiving their beacons. This operation can be done while being available on the advertised frequency bands or while sensing the spectrum. The information about surrounding nodes should be stored in a database containing the latest-received beacons of every CR. Since the characteristics of the hopping pattern of surrounding nodes is not known, the best approach to discover other CRs is to randomly hop in the frequency spectrum. With our proposition, CRs can thus rendez-vous with unknown nodes at no cost and can perform sensing if and when wanted.

D. Updating the hopping pattern

If two CRs need to communicate often or have a low-latency requirement, they need to adjust their hopping pattern so as to maximise the time they spend being reachable. Changing the hopping pattern of a node can be challenging to do locally without breaking the connectivity of the CRN. It is safe to update the frequency hopping period or drop a frequency band as long as every node in the node's neighbours CR list will still be available. This can be checked by finding at least one periodical overlap between the node's new beacon and every other beacon from nodes it can already communicate with. It is however unsafe to update the beacon and to rely on a neighbour CR to maintain the connectivity because they may change their hopping pattern in the future.

Since our PHY-layer signalling protocol provided the discovery and the loose time synchronisation necessary for communicating with surrounding nodes and since the connectivity problem cannot be addressed entirely locally, we believe that optimising the hopping pattern is outside the scope of our proposed PHY-layer signalling protocol and that it should be addressed at the network layer, with a cross-layer approach.

E. Evaluation

To evaluate our proposition, we wrote a simulation in C++ with two CRs with a tunable band ranging from 300MHz to 3GHz and a spectrum window of 25MHz.

The first node has a hopping pattern period of 1s and advertises two bands that are available respectively in [0.0, 0.4] and [0.5, 0.9]. The two bands are selected randomly at the beginning of the experiment. The node will send a beacon 5ms after switching to a band and at a user-defined period after that. Beacons are sent at 1 MBit/s, using a simulated bandwidth of 500kHz and 280μs to send it, according to the binary structure of the beacon found in Figure 2.

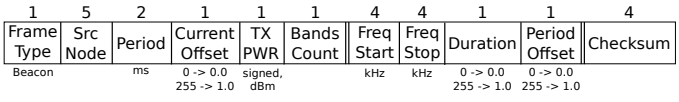

Figure 2: Format of the beacon frame

The second node performs only sensing. It randomly hops from one band to another with a user-defined hopping period. The experiment finishes when the sensing node is able to hear the entirety of a beacon sent by the first node.

Figure 3 shows the average time (over 1000 instances) it takes for the sensing node to receive a beacon from the other node depending on both the beaconing period and the sensing hopping period. In "shp = 1ms", the sensing CR has

a high probability of not hearing the complete beacon because the sensing period is short compared to the emission time (280μs) of the beacon. Multiplying by 10 the beaconing period increases the average rendez-vous 10 folds until reaching one second. After this point, the only beacon sent by the CR is the one sent when hopping to a new band because the hopping cycle takes 1 second and no increase in the delay can be observed. The same behaviour can be observed with the other sensing hopping periods with an added initial plateau due to the sensing hopping period being higher or equal to the beaconing period, ensuring reception of the beacon. The sensing period should thus be set around 10ms and the beaconing period set as low as possible (based on the utilisation of the band) to achieve the lowest discovery delay possible. The source code of the simulation is available at https://github.com/mupuf/hachoir.

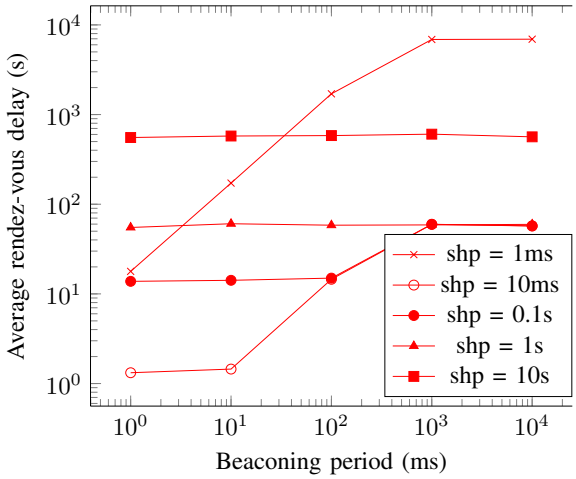

Figure 3: Influence of the beaconing and sensing-hopping period on the average rendez-vous delay.

V. MAC-layer signalling protocol

The role of our MAC-layer signalling protocol is to provide a handshake mechanism that allows unicast communication by selecting the PHY-layer parameters that should be used to transmit a frame. Contrarily to the PHY-layer signalling protocol which advertises the generally-available frequency bands, the MAC-layer signalling protocol selects a frequency band that is available at the time of transmission and picks a modulation that is compatible with the channel fading and the time available before one of the two nodes jumps to another frequency band. Our proposition is based on the "Willing to Send" (WTS), "Ready To Receive" (RTR) and "Ready to Send" (RTS) control messages, shown in Figure 4.

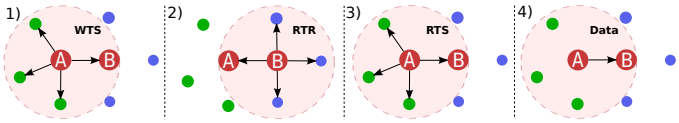

Figure 4: Overview of the MAC signalling protocol

A. The WTS, RTR and RTS frames

```
<WTS_frame>{ src=23, dst=12, data_len = 589,
  [ {band1}, {band2}, {band3} ]
  [ {modulation1}, {modulation2} ]
  deadline=152ms, expires=20ms, tx_pwr=20dBm }
```

In this example, the WTS frame indicates that node *23* wants to send *589* bytes to node *12*. The transmission can happen in any sub-band of *band 1, 2 or 3* and should use *modulation1* or *modulation2*. A band is composed of a central frequency, a bandwidth and the maximum transmission power that can be used without perturbating any neighbouring primary user. A modulation is composed of its type (BPSK, QAM16, ...) and the maximal symbol rate that can be sent. The receiver has *152 ms* to receive the message, starting from the moment the WTS frame got emitted. After this point, the emitter will jump to another frequency band. Every other node receiving this WTS frame should refrain from emitting or accepting new transmissions in these frequency bands, during at least *20ms* or until a follow-up RTR or RTS frame is received. This parameter should be set according to the maximum time node 12 is supposed to take before answering back to node 23 so as, if node 23 is not able to decode this transmission, the lock on the frequency band can be lifted earlier than 152ms. The WTS frame is sent simultaneously on bands 1, 2 and 3 with a transmission power of *20dBm* which will allow node 12 to assess the fading found at every frequency band and request the wanted TX power from node 23 in order to have a sufficient SNR to reach the fastest modulation possible. The WTS, RTR and RTS frames should use a modulation that supports poor SNRs to increase the chance of reception in the surrounding nodes at the expense of transmission time of the control frames and the general decrease in throughput.

Upon reception of the WTS frame, node 12 uses the content of the Radio Event Table defined in III-B and shown in Figure 1 in order to select the best candidate bands that intersect with the available bands found at node 23 and that have not been reserved by another node's WTS, CTS or RTS frame. Based on the fading found on the selected band (5dBm - received power before amplification) and the maximum emission power of node 23 for each selected band, node 12 selects the bands with the highest SNRs. Node 12 then selects the modulation (type and symbol rate) which: (i) is compatible with its and the transmiter's capabilities, (ii) fits in the frequency band selected, (iii) works with the expected SNR and (iv) allows transmitting the 589 bytes fast-enough to meet the deadline. If a solution that meets all the criterias is found, then node 12 can emit on the selected band the following RTR frame, containing the PHY parameters (selected band, modulation type, symbol rate and transmission power) that node 23 should use to transmit its frame. Nodes receiving this frame can now cancel all the reservation on all the frequency bands selected by the previous WTS frame and only lock the band found in the RTR frame for the next 23ms. This delay is the expected time it will take for the emitting node to handle the RTR frame, emit the RTS frame and send the data frame at the expected data rate.

```
<RTR_frame>{ src=12, dst=23, data_len = 589,
  {band}, {modulation}, tx_pwr=15dBm,
  expires=23ms }
```

When receiving the RTR frame, node 23 should immediatly emit an RTS frame on the selected band containing the frequency band that will be used for the transmission to node

12 along with the time during which the surrounding nodes should refrain from using it. The RTR frame takes precedence onto the previous RTR and WTS frames. The expiration time is lower than the RTR frame because it takes time for the samples to reach the software that will demodulate the signal and decode the frame.

```
<RTS_frame>{ src=23, dst=12, {band},
  tx_pwr=15dBm, expires=19ms }
```

Node 23 now has 19ms to send the data using the PHY-layer parameters that were defined in the RTR frame. The 4ms difference with the RTR's expiration time is an example of the processing time needed for the samples to reach the signal processing software, being demodulated, decoded, sent to the network stack and vice versa for the emission of the RTS frame.

The WTS/RTR/RTS frames' binary representation is shown in Figure 5. The minimum size of a WTS frame is 36 bytes, with an additional 8 bytes per added bands and 4 bytes per added modulation. The size of the RTR and RTS frames is respectively 32 and 28 bytes.

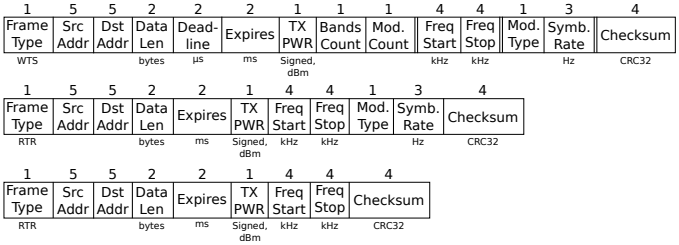

Figure 5: Format of the MAC frames (WTS, RTR then RTS)

B. Discussion

Our MAC-layer signalling protocol is, in its nature, an enhancement of the IEEE 802.11 RTS/CTS handshake. It thus inherits its characteristics among which is to partially prevent the hidden node problem. However, since we are using this mechanism in a CRN environment, nodes may hop in and out of the frequency band selected during the transmission of a frame. If a node hops in the frequency band after the RTR frame is sent, is far-enough from the emitter not to be able to sense the transmission and yet is close enough to perturbate the receiver in case it decides to send data on this band, it may create a collision. One way to mitigate this problem is to have a relatively low hopping frequency compared to the time it takes to send a frame and to mandate that nodes should be mute for a few ms when hopping on a new band.

Aside from the noted limitation and the increased size of the frames, our proposition should behave in the exact same way as the IEEE 802.11 RTS/CTS handshake and the same limitations and mitigations. This means that when the risks of collisions are low or when the message to be sent is small, this mechanism increases latency and decreases throughput needlessly and can be bypassed.

VI. Conclusion & future work

In this article, we proposed a sensing technique for detecting transmissions from primary and secondary users in order to identify available frequency bands.

We then proposed a discovery mechanism that allows nodes to temporally and spatially synchronise to enable communication among CRs. This mechanism is a clear improvement over the state of the art because it allows guaranteeing availability on a number of bands while allowing nodes to perform sensing on other bands or saving power by powering-off their radios. In our simulations, the synchronisation could happen in as little as 1.32s in average for a realistic scenario.

Our MAC-layer signalling protocol proposition brings advanced cognitive behaviours to frame transmissions by allowing to pick the least-crowded frequency band, the weakest transmission power and the fastest modulation possible to reach the destination while creating as little perturbations as possible on surrounding primary or secondary users.

Future work will focus on dynamic reconfiguration of the hopping pattern to modulate the availability and latency between one or multiple nodes according to the wanted QoS. Support for hardware radios is also partially possible and will be investigated. We will also investigate the possibility for the MAC-layer signalling protocol to specify a frequency-hopping pattern or parallel transmission in the RTR and RTS frames in order to increase the throughput in heavily-fragmented frequency bands. Finally and more importantly, we will propose extensions to add support for broadcast and multicast capabilities in our architecture without affecting the current performance of our CRN.

Acknowledgment

Part of this work has been funded by the Leveraging Insurance for services providers cohabitation over Cognitive Radio Networks (LICoRNe) project from the French National Research Agency (ANR). We also would like to thank Vincent Rose for finding the noise's model on our USRPs.

References

[1] C.-S. Hsu, Y.-S. Chen, and C.-E. He, "An efficient dynamic adjusting MAC protocol for multichannel cognitive wireless networks," in *2010 IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS)*, Jun. 2010, pp. 556–560.

[2] A. De Domenico, E. Strinati, and M. Di Benedetto, "A survey on MAC strategies for cognitive radio networks," *IEEE Communications Surveys Tutorials*, vol. 14, no. 1, pp. 21–44, 2012.

[3] Y.-C. Liang, K.-C. Chen, G. Li, and P. Mahonen, "Cognitive radio networking and communications: an overview," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 3386–3407, Sep. 2011.

[4] Z. Lin, H. Liu, X. Chu, and Y.-W. Leung, "Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks," in *2011 Proceedings IEEE INFOCOM*, Apr. 2011, pp. 2444–2452.

[5] R. Doost-Mohammady, P. Paweczak, G. Janssen, and H. Segers, "Physical layer bootstrapping protocol for cognitive radio networks," in *2010 7th IEEE Consumer Communications and Networking Conference (CCNC)*, Jan. 2010, pp. 1–5.

[6] G. Nychis, T. Hottelier, Z. Yang, S. Seshan, and P. Steenkiste, "Enabling MAC protocol implementations on software-defined radios," in *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, ser. NSDI'09. Berkeley, CA, USA: USENIX Association, 2009, pp. 91–105.

[7] A. P.S and M. Jayasheela, "Cyclostationary feature detection in cognitive radio using different modulation schemes," *International Journal of Computer Applications*, vol. 47, no. 21, pp. 12–16, Jun. 2012.