Ad Hoc Networks 8 (2010) 430–438

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Common control channel design for cognitive radio wireless ad hoc networks using adaptive frequency hopping

Claudia Cormio [a], Kaushik R. Chowdhury [b,*]

[a] Telematics Lab., Dipartimento di Elettronica ed Elettrotecnica, Politecnico di Bari, Italy
[b] School of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, United States

ARTICLE INFO

Article history:
Received 21 September 2009
Accepted 16 October 2009
Available online 3 November 2009

Keywords:
Cognitive radio
Opportunistic spectrum access
Common control channel
Media access control

ABSTRACT

Cognitive radio (CR) technology enables the opportunistic use of the portions of the licensed spectrum by the CR users, while ensuring low interference to the primary user (PU) activity in the licensed bands. The spectrum is sensed locally by the CR users, and a specific channel that is acceptable to both the end nodes of the communication link is chosen. However, this necessitates a common control channel (CCC) for exchanging the sensing information and reserving the channel before actual data transfer. In this paper, a common control channel design for CR ad hoc networks is proposed, called as adaptive multiple rendezvous control channel (AMRCC) based on frequency hopping. Our scheme is scalable, and allows continuous connectivity between the CR users under dynamic PU activity. The contribution made in this paper is threefold: (i) a frequency hopping scheme is proposed that allows altering the hopping sequence based on the PU activity in the channels, (ii) a simple and low-overhead procedure is developed to aid new node-join and leave events, and (iii) a slot duration optimization is given that avoids a significant performance degradation with the number of available channels. Performance evaluation proves that our solution achieves better performance than the other classic CCC solutions in terms of time to rendezvous (TTR) and the resulting throughput, specifically in CR ad hoc networks.

1. Introduction

Cognitive radio networks allow the CR users to share the wireless channel with licensed or primary users (PU) of the spectrum in an opportunistic manner. Such a radio is equipped with dynamic spectrum access capability, thereby allowing it so identify portions of the spectrum that are currently available for transmission. This technology is envisaged to solve the problem of spectrum scarcity in the unlicensed 2.4 GHz ISM bands, and the inefficient spectrum utilization that exists in some licensed frequency bands.

One of the key considerations in using CR technology is ensuring that the PU transmission is always protected. To achieve this, it is essential to exchange the information pertaining to the spectrum availability between a given node pair before starting data transmission. This functionality is generally provided at the link layer, and several works have addressed the problems of medium access control (MAC) for CR networks [6]. The MAC protocols assume either a dedicated control channel with reserved frequencies within the licensed band, or the explicit use of the unlicensed band to exchange the control information [1]. These messages exchanges over the common control channel (CCC) may be related to the (i) channel access and contention, (ii) neighbor discovery, and (iii) spectrum management. While the first two approaches are common to classical wireless ad hoc networks, there are several spectrum related functions that are unique to CR networks. Specifically, the spectrum sensing information has to be exchanged over a channel that is always available before data can be transmitted. Moreover, once a PU reclaims

* Corresponding author. Tel.: +1 404 510 9089.
E-mail addresses: c.cormio@poliba.it (C. Cormio), krc@ece.neu.edu (K.R. Chowdhury).

1570-8705/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2009.10.004

the spectrum, the new channel must be quickly determined for the affected link, and this link recovery information cannot be transmitted over the previously used spectrum. In either case, a CCC is used to facilitate the continuous operation of the CR users without any disruption.

Most of the current works implicitly assume the presence of such a control channel and hence, formalizing such a CCC design idea is very important. In this work, we propose a frequency shifting design for a CCC, called as AMRCC that can adaptively change the sequence of the channels based on the sensing results. Our design is motivated by the fact that strict synchronization is difficult to achieve in an ad hoc network, and the CCC design must be scalable. Moreover, we assume a single radio network, where the CR users may be mobile and operate under the standard concern of minimizing the interference to the active licensed users in the vicinity.

The research community has proposed several schemes for setting up and maintaining a reliable CCC in CR networks, while addressing the challenges posed by (i) saturation of the channel reserved for the CCC [9], (ii) robustness to PU activity [1], (iii) jamming attack [10], and (iv) limited control channel coverage [1]. We describe some of the related works in the following discussion.

A swarm intelligence based dynamic control channel assignment is proposed in CogMesh [4]. It is a distributed and adaptive approach, which addresses the issue of the PU activity disrupting the use of the CCC, and the limited coverage of the control channel. This is the first work that applies a swarm intelligence-based algorithm, where higher efficiency is achieved if specialized workers perform specialized tasks in parallel. In response to the detected PU activity, the best channel chosen by the majority of the CR users is used as the CCC. Furthermore, the coverage is improved by reducing the number of control channels in the network, thereby decreasing overhead and switching delay. The maintenance of the proposed CCC incurs high overhead, especially in the presence of fast varying PU activity. Moreover, it assumes precise synchronization among the CR users.

The SYN–MAC is a slotted protocol that integrates the control channel access with the regular available data channel [8]. Each time slot is dedicated to a channel, which can be used both for control and data exchange. This also addresses the problems of control channel saturation and jamming. The main drawbacks here is that it disrupts the control channel operation whenever a PU occupies the control slot. The SRAC protocol addresses the control channel issue by exploiting the techniques of dynamic channelization and cross-channel communication [10]. The first technique refers to the capability to adapt the band for control signaling in order to prevent saturation, while the second allows a node to dynamically change the receiving channel in order to avoid jamming. However, this work is limited from the viewpoint of network-wide CCC coverage, and cannot easily recover from a sudden PU appearance.

The C-MAC protocol is characterized by in-band signaling [5]. It guarantees flexible control channel coverage, and it is robust to PU activity through the use of backup channels. The main drawbacks of this work are the high control overhead due to beacon exchange and the requirement of strict synchronization. The necessity for synchronization are more relevant to networks with special topologies, such as networks based on *clusters*. Such a solution is presented in [2], where the clusterhead assists in the choice of the channel. Moreover, several different clusters representing different CCCs may be integrated over time. This work may is limited in application as it assumes a special topology, and is also adversely affected in dynamically changing node locations.

A multiple rendezvous control channel (MRCC) is proposed in [7]. Here, the nodes hop over multiple channels till a common channel to the pair (rendezvous channel) is reached. The main problem of this work is that the hopping sequences between a node pair is static once the rendezvous condition is reached. If the PU activity is detected on a channel, the hopping on that particular channel is simply removed from the schedule. For dynamically changing PU activity, this may lead to inefficient hopping schedules. Moreover, there is no bounded time to achieve the rendezvous (TTR).

The adaptive multiple rendezvous control channel (AMRCC) scheme proposed in this paper maximally spreads control signalling and data transmission among channels by overcoming the limits of [7]. In fact, the AMRCC scheme builds the channel hopping sequences based on sensing information in order to hop across the different channels by minimizing the interference to licensed users. The main improvement of our work over [7] is that the sequences are chosen adaptively to combat the problem of PU interference. Similar to Bluetooth [3], the hopping sequences are built in such a way that the channels with minimum interference to other devices occur a higher number of times than the others. As opposed to this, [7] uses pre-defined sequences based on permutations of the available channels. The PU activity is taken into account by just removing a channel from the pre-defined sequence. This raises the concern that the sequences get progressively shorter. Additionally, AMRCC is completely asynchronous, unlike [7], where the start and stop times of each slot have to be rigidly synchronized.

An important contribution of this paper is a slot duration optimization of the AMRCC scheme. Here, the time for which the CR user stays on a given channel is also variable, and a function of the PU activity on the channel.

The rest of the paper is organized as follows: Section 2 the motivates our work. The design of our proposed AMRCC control channel design is explained, together with an optimization which assures scalability with the number of channels in Section 3. An optimization to the AMRCC design with differential slot times is described in Section 4. Section 5 gives a detailed performance evaluation of the AMRCC scheme and, finally, we conclude in Section 6.

2. Motivation for an adaptive MRCC

In this section we propose our adaptive MRCC (AMRCC) scheme for CR ad hoc networks, which achieves high performance by dynamically adapting the hopping sequences to the detected PU activity.

We chose to extend the MRCC solution for our approach because it may be used to maximally spread the control and data packets among the channels of the licensed spectrum, thereby making the CCC robust to the unpredictable PU activity. Furthermore, it guarantees high throughput by exploiting spectrum holes in an efficient way with minimum delay.

Classic multiple rendezvous control channel schemes have several drawbacks which make them infeasible in cognitive radio networks. They are (i) strict synchronization, (ii) low scalability, and (iii) low robustness to PU activity changes. We explain these factors as follows:

- *Strict synchronization:* Most hopping sequences require the sender-receiver pair to hop together, in which the precise start and end times of the individual hops must be synchronized at all times. This is difficult to achieve and maintain in distributed environments. In AMRCC, the channel switching times may not be the same for all the users during the network operation. The synchronization is only enforced after the rendezvous, i.e. after both the sender and the receiver meet on a common channel during the hopping process. This is when the SYNC packet is exchanged between the users. As opposed to this, the classical MRCC approach requires a strict synchronization, starting right from the network initialization phase.
- *Low scalability:* In the classical wireless networks, every time a new node appears, it should exchange the sequence seed with all its neighbors. This action is necessary in order to be able to follow the hopping pattern of the intended receiver in the neighborhood. Our adaptive scheme, on the contrary, results in a significantly reduced overhead as the seed exchange only occurs at the rendezvous channel. The sender and its intended receiver hop in an independent and asynchronous manner over their own sequences, which are completely unknown to each other before the rendezvous. Thus, this requires fewer timer settings and schedule maintenance operations as compared to the classical MRCC.
- *Low robustness to PU activity changes:* The classical MRCC solutions are not suitable for cognitive radio networks as they are not able to adapt to the dynamic behavior of the primary network and, consequently, avoid interference to the PUs. Previously proposed solutions assume that after the rendezvous of a node pair over a given channel, they perform the entire data packet exchange over that channel. There is no mechanism to vacate the spectrum band due to the appearance of a PU. In our approach, the hopping sequence is modified on the basis of the latest sensing results by favoring the data exchange over the channels with the lowest PU activity.

We now describe our proposed approach in detail in the following section.

3. AMRCC scheme description

In this section, we introduce our adaptive MRCC (AMRCC) scheme, which aims at improving the time to rendezvous (TTR) and the overall network performance by overcoming the performance issues of the classical MRCCs discussed in Section 2. The goal of our approach is to achieve a lower time to rendezvous and higher throughput, which means better exploitation of the spectrum holes in the PU network, compared to the classical solutions.

3.1. Overview

In Fig. 1 the entire protocol behavior within a period of the sensing cycle is shown. A node performs spectrum sensing periodically after a time out T_{out} and the period of the sensing cycle is assumed to be equal to the sum of the sensing duration and the time out period. The sensing results are used to build a ranking table of the available channels based on the PU activity detected on each channel. The node then generates a pseudo-random hopping sequence, which is mapped to the ranking table in order to build an adaptive hopping sequence. The frequency hopping performed over the adaptive hopping sequence increases the probability to achieve rendezvous on a channel with low PU activity, and to decrease the time to attain the rendezvous. When the rendezvous is achieved, sender and receiver synchronize by exchanging SYNC packets. The SYNC packet includes the information for building the common hopping sequence. The two nodes then start hopping on the sequence of common channels in order to exchange data packets until the expiration of the time out period, following which a new round of sensing is performed.

The different phases within a period of the sensing cycle can be summarized by the following routines: (i) sensing, and (ii) handshaking. We describe these phases as follows:

3.2. Sensing routine

It is performed periodically with a period equal to the sum of the sensing duration and the time out T_{out} interval. This routine gives the adaptive hopping sequence based on the sensing results. At the start up time, the CR users asynchronously begin their activity by performing spectrum sensing. As no network-wide silence duration is enforced, the CR user may not be able to distinguish between a signal originating from a PU, from those caused by other CR users, if simple energy detection techniques are used. For this reason, we assume that a feature detection scheme is in place [1]. After the sensing is completed, the node builds

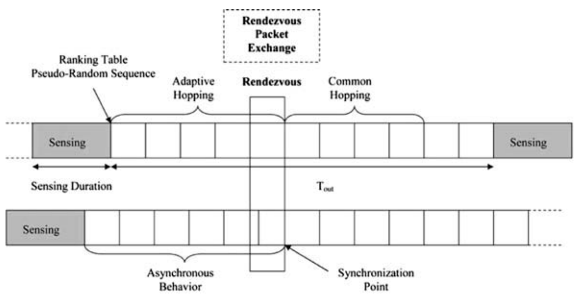

Fig. 1. Overview of the protocol behavior.

a channel *ranking table* based on the sensing results. In particular, a channel ranking table is a table where channels are ordered based on the PU activity, starting from the channels where the lowest PU activity is detected. Note that neighboring nodes experience, with high probability, a similar PU activity across the channels, implying high correlation among their channel ranking tables and, consequently, between their adaptive hopping sequences.

After building the channel ranking table, the node generates a *pseudo-random sequence* containing integer values in the interval [1, c], where c is the number of channels. The pseudo-random sequence is biased towards the lowest values, i.e. the lower the value the more often it occurs in the sequence. We propose two variants of AMRCC called as the (i) basic AMRCC scheme, and the (ii) enhanced AMRCC scheme by either using a decreasing linear function, or a parabolic decreasing function, respectively. In the basic AMRCC, the number of occurrences y of each value x in the sequence is given by,

$$y = -b \cdot x + c + 1, \tag{1}$$

where b is the slope of the line and c is the number of channels. For simplicity we assume b to be 1. In the enhanced AMRCC scheme, the number of occurrences y of each value x is given by,

$$y = (c - x + 1)^2, \tag{2}$$

where c is the number of channels. We explain the simple mechanism of determining the number of occurrences of each value in the sequence in the AMRCC, assuming $ch = 5$ channels available. Thus, the sequence contains integer values from the interval [1,5]. If a decreasing linear function is used (with reference to the basic AMRCC) the value 1 occurs 5 times in the sequence, the value 2 occurs 4 times, and so on. If a decreasing parabolic function is use (with reference to the enhanced AMRCC) the value 1 occurs 25 times in the sequence, the value 2 occurs 16 times, the value 3 occurs 9 times, and so on. The total length l of the sequence for the basic AMRCC is given by the Gaussian formula:

$$l = c(c + 1)/2, \tag{3}$$

where c is the number of channels. In the enhanced scheme, the length l of the sequence is given by,

$$l = \sum_{i=1}^{c} (c - i + 1)^2, \tag{4}$$

where c is again the number of channels. In both cases the sequence is built by sampling without replacement the values from a set I, where the set I is defined as the set of the elements $i \in N$, where the number of occurrences of each element i is equal to y_i. The set I is defined as follows:

$$I = \{i | i \in N|\}, \tag{5}$$

where $N = \{1 \dots c\}$. The number of occurrences y_i is given for the basic AMRCC scheme by,

$$y_i = -i + c + 1, \tag{6}$$

and for the enhanced AMRCC scheme by,

$$y_i = (c - i + 1)^2, \tag{7}$$

where c is the number of channels. In both the cases, the sequence is repeated in a periodic manner until a change is forced by the protocol, such as the case when a sensing operation is performed or the rendezvous is achieved. The following step is the *mapping* between the pseudo-random sequence and the channel ranking table. In particular, value i in the sequence is substituted with the ith channel in the ranking table. In this way, the channels with higher ranking occur more often in the resulting sequence, which makes it extremely adaptive to the detected PU activity. A similar concept is exploited by Bluetooth devices as explained in [11], where the hopping sequences are altered in order to avoid channels with the highest interference. An overview of the sensing routine, giving as an output the adaptive hopping sequence, is given in Fig. 2.

3.3. Handshaking routine

This phase begins when a node, which is hopping over the adaptive hopping sequence, has a packet to transmit. The handshaking procedure gives as an output the common hopping sequence after the rendezvous between the node and the intended receiver is achieved. After the sequence has been computed, the node starts hopping over the adaptive hopping sequence. As soon as a new packet arrives from the upper layers, the node starts sending a request to send (RTS) packet on each channel it hops in. If the intended receiver is on the same channel that the RTS packet is sent, it replies with a clear to send (CTS) packet. Upon receiving the CTS from the intended receiver, the *rendezvous* procedure is completed. The time interval between the instants at which the node starts the handshake, and that in which it arrives at the rendezvous channel is referred as time to rendezvous (TTR). Now, the sender and receiver exchange SYNC packets in order to synchronize the future hops, and then exchange the *rendezvous packet*

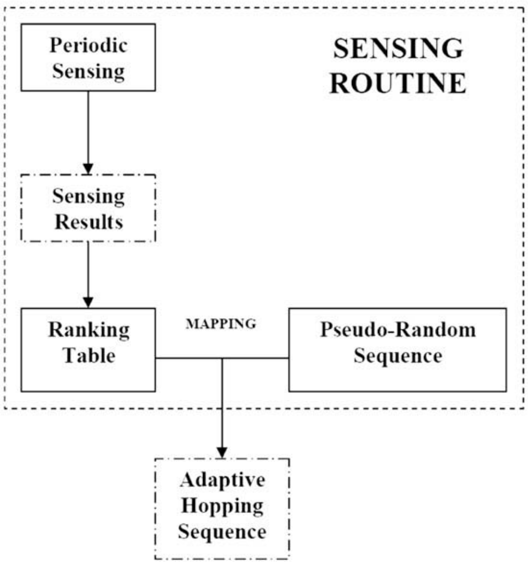

Fig. 2. Overview of sensing routine.

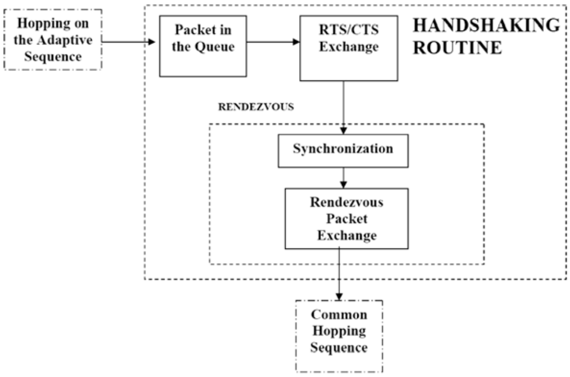

Fig. 3. Overview of handshaking routine.

that contains their ranking tables, the seed of their pseudo-random sequences, and the time elapsed from the last sensing event, namely T_{ls}.

The next step for the node and the intended receiver is to compute the *common hopping sequence* they will both hop on in order to exchange data packets. The adaptive hopping sequence is related to the lowest value of T_{ls} between sender and receiver. This sequence is assumed by both the nodes as the common hopping sequence which is kept updated by following the above procedure. The nodes can easily compute each other's sequence from the information included in the rendezvous packet. An overview of the handshaking routine, which produces as an output the common hopping sequence, is given in Fig. 3. After exchanging data while hopping on the common hopping sequence, the transmitter and receiver keep hopping till a new sensing action is performed, and a new adaptive hopping sequence is computed. Note that unlike the most classic MRCCs [7] our solution does not cause high overhead in order to configure nodes joining the network, as seeds and ranking tables are exchanged only *after* rendezvous, thereby ensuring scalability. As opposed to this, in previous solutions, when a new node joins the network, it must be configured by receiving the seed of other nodes' pseudo random sequence in order to start hopping on the same sequence of its intended receiver.

4. AMRCC scheme with variable slots

In this section, an optimization for the AMRCC is presented that provides greater adaptation to the PU activity. We recall that the AMRCC already takes into account the issue of the PU activity by using hopping sequences. In this optimization, we allow the channels with less PU activity to occur a higher number of times within a single transmission frame as compared to the other channels.

By allowing a CR user to occupy the infrequently used licensed channels longer (by having more slots dedicated to such channels and increasing the slot duration), the probability of affecting the PU is further reduced. We explain the concept of the variable slot as follows:

In both in the basic and the enhanced adaptive hopping sequence, the duration of the slot s, related to the channel with the highest ranking $r = 1$, is equal to,

$$s = c \cdot u, \tag{8}$$

where c is the number of channels and u is the slot duration unit. The slot duration unit corresponds to the duration of the slot for the channels with ranking $r > 1$ (see Fig. 4). Consider an example with $c = 5$. The duration s of the slot for the channel with rank $r = 1$ is equal to 5 times the slot duration unit u, while the slots related to all the other channels with ranking $r > 1$ have duration equal to u. By applying this optimization to the basic AMRCC, a given user stays on the channel with the highest ranking, during one frame of the sequence, for the time $c^2 \cdot u$, considering that each slot for the channel with $r = 1$ is equal to $c \cdot u$ (Eq. (8)). Also, the number of slots is equal to c with Eq. (1) where $b = 1$ and substituting $x = r = 1$. The global duration of the sequence is obtained by summing (i) the time $c^2 \cdot u$ that a node stays on the channel with the highest ranking, and (ii) the time a node stays on all the other channels with ranking $r > 1$. The second term of the addition is equal to $\frac{(c-1)\cdot c}{2} \cdot u$, computed by using Eq. (3), by replacing c with $c - 1$, because $c - 1$ is the number of channel with ranking $r > 1$. Consequently, the probability p that a node stays on the channel with the highest ranking within one sequence frame in the basic AMRCC with variable slots is given by,

$$p = \frac{c^2 \cdot u}{c^2 \cdot u + \frac{(c-1)\cdot c}{2} \cdot u} = \frac{2 \cdot c}{3 \cdot c + 1}. \tag{9}$$

In the basic AMRCC with fixed slots, p is given by,

$$p = \frac{c \cdot u}{\frac{(c+1)\cdot c}{2} \cdot u} = \frac{2}{c + 1}, \tag{10}$$

obtained considering that the time a node stays on the channel with the highest ranking channel is equal to $c \cdot u$, and the global duration of the sequence is equal to $\frac{c(c+1)}{2} \cdot u$ by using Eq. (3).

Analogous equations can be obtained for computing the probability p that a node stays on the highest ranking channel in the enhanced AMRCC with variable and fixed slots. The probability p, in case of variable slots, can be derived considering that a given node stays on the channel with the highest ranking, during one frame of the sequence, for the time $c^3 \cdot u$. Here, each slot related to that channel is equal to $c \cdot u$ (Eq. (8)), and the number of slots is equal to c^2 (Eq. (2)), where $x = r = 1$. The global duration of the sequence is then obtained as the sum of the time $c^2 \cdot u$ a node stays on the channel with the highest ranking, and the time a node stays on all the other channels with ranking $r > 1$. The second term can be computed using Eq. (4) by summing $c - 1$ terms instead of c, as we already considered the term with $i = 1$ related to the channel with

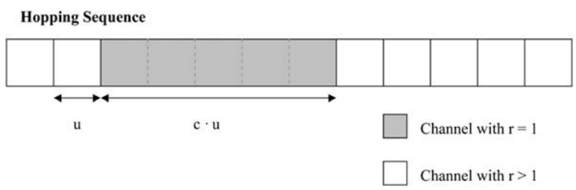

Fig. 4. Variable slot duration in the hopping sequence.

highest ranking. This term is given by $\sum_{i=1}^{c-1}(c-i+1)^2 \cdot u$. Consequently, the probability p that a node stays on the channel with the highest ranking in the enhanced AMRCC with variable slots is given by,

$$p = \frac{c^3 \cdot u}{c^3 \cdot u + \sum_{i=2}^{c}(c-i+1)^2 \cdot u} = \frac{c^3}{c^3 + \sum_{i=2}^{c}(c-i+1)^2}, \tag{11}$$

which can be written in closed form, using the Heron's Formula, as,

$$p = \frac{6c^2}{8c^2 - 3c + 1}. \tag{12}$$

In the enhanced AMRCC with fixed slots, p is given by,

$$p = \frac{c^2 \cdot u}{\sum_{i=1}^{c}(c-i+1)^2 \cdot u} = \frac{c^2}{\sum_{i=1}^{c}(c-i+1)^2}, \tag{13}$$

obtained considering that the time that a node stays on the channel with the highest ranking is equal to $c^2 \cdot u$ and the global duration of the sequence is equal to $\sum_{i=1}^{c}(c-i+1)^2 \cdot u$ by using Eq. (4). It can be written in closed form as,

$$p = \frac{6c}{2c^2 + 3c + 1}. \tag{14}$$

Fig. 5 shows that, with the solutions with fixed slots, the probability p decreases with the number of channels c, as the length of the sequence also increases, leading to a degradation of performance linked to longer time to rendezvous. By implementing the AMRCC with variable slots instead, the disadvantage caused by longer sequences when c increases is compensated by the longer slot related to the channel with the highest ranking. The probability p, in this case, is nearly constant, implying that the global performance does not undergo a significant degradation when c is significantly high.

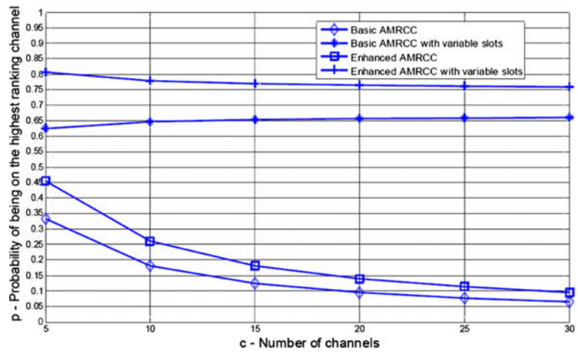

Fig. 5. Percentage of time occupied by the highest ranking channel in the adaptive sequence.

4.1. Analysis of the AMRCC scheme with variable slots

Let λ_i be the expected number of PU occurrences on a channel i in a slot time u, each slot being of the same length before the slot duration optimization. After the optimization, however, the slot duration for the rank 1 channel is extended, equal to $c \cdot u$. The expected number of occurrences now in this preferred channel, say x, is $c \cdot \lambda_x$.

Assuming the arrival distribution to be Poisson, the probability that there are no PU arrivals in a given slot in channel i with rank $r > 1$ is,

$$p_i = e^{-\lambda_i}. \tag{15}$$

In particular for the channel x with rank $r = 1$, the probability of no PU arrivals is,

$$p_x = e^{-c \cdot \lambda_x}. \tag{16}$$

Thus, the probability of interference in at least one channel (or slot) is given by,

$$P = 1 - p_x \cdot \prod_{j=2}^{k} p_{ch_j}, \quad ch_j \neq x, \tag{17}$$

where k is the number of slots in the frame, and ch_j is the channel used in the jth slot. This can be simplified as,

$$P = 1 - e^{c \cdot \lambda_x + \sum_{j=2}^{k} \lambda_{ch_j}}, \quad ch_j \neq x. \tag{18}$$

This probability is considerably smaller than the case of AMRCC with fixed slots given by,

$$P = 1 - e^{\sum_{j=1}^{k+c} \lambda_{ch_j}}, \tag{19}$$

as $\lambda_x \ll \lambda_i, \ i \neq x$. In the following section the performance evaluation of our scheme is given for both the basic and the enhanced approaches while considering the slot duration optimization explained in this section. We compare our our work to the related MRCC solution [7].

5. Performance evaluation

5.1. Simulation setup

A simulation tool in Matlab was built in order to evaluate the performance of the AMRCC, focusing in particular on the schemes with variable slots. We assumed a single hop scenario where all the n CR users are in the same radio range. This assumption implies that all the nodes detect the same PU activity during their sensing periods, and thus, produce correlated pseudo-random sequences which lead to lower time to rendezvous (TTR) and also result in better global performance. The number of available channels c is set in the beginning and does not change during the simulation time Δ_{ob}. The traffic for both the PU and CR user has a Poisson distribution with mean value for the packet waiting time respectively equal to λ_{PU} and λ_{CR}. The default behavior of CR users is asynchronous and it is implemented by introducing, at the network initialization, a random delay offset bounded at the value off_{max}. The users can synchronize among themselves, during the simulation, at the rendezvous point in order to exchange data packets. Each user starts its activity by performing sensing and a fixed sensing duration Δ_s has been set for all the nodes. The user performs sensing by picking samples of the PU Poisson Traffic within its sensing period. As the sensing periods of all the nodes are not overlapping, the ranking tables may be slightly different, although the PU activity is the same for all the users in the same range. The same time out $T_{out,s}$ for sensing is set for all the CR users. Moreover, the same duration Δ_h for the basic time slot is set for all the users. The duration of the slot related to the channel with the highest ranking is a multiple of the basic slot duration depending on the total number c of available channels.

If a packet arrives in the queue, according to the Poisson traffic, while the user is performing sensing or handshaking, it is enqueued and waits untill the rendezvous after which the queue can be emptied.

In this analysis, we have ignored the collisions generated by the control packets sent by CR users during the handshake. In fact, a single slot has a large number of RTS/CTS messages. This implies that, even if some RTS/CTS packets collide among themselves or with PU packets within a slot, the number of the RTS/CTS packets globally sent in the slot is sufficient to always enable a successful rendezvous, if sender and receiver are over the same channel.

Moreover, for simplicity, it is assumed that, in case of collision between a CR user packet and a PU packet, the CR user packet is dropped instead of being retransmitted after a backoff. Finally, it is assumed to ignore collisions among CR user packets, because the goal of the paper is to show the CR user behavior towards the PU activity, putting in evidence how efficiently CR users are able to exploit the spectrum holes.

5.2. Simulation results

First, some results for the time to rendezvous (TTR) and the average throughput of CR users by using the basic and enhanced AMRCC with slot duration optimization are given, together with a comparison with a previous sequence based MRCC solution [7].

The main parameters set in the simulations are defined as follows: the global observation time $\Delta_{ob} = 15,000$ s, the duration of one hop $\Delta_h = 5$ s, the sensing duration $\Delta_s = 15$ s, the time out for sensing $T_{out,s} = 500$ s, the mean waiting time for PU traffic $\lambda_{PU} \in [5, 90]$ s, the mean waiting time for CR traffic $\lambda_{CR} = 5$ s, the number of CR users $n = 10$, the maximum offset for the CR user clock $off_{max} = 5$ s, and the number of channels $c \in [5, 30]$. The results shown in this section are averaged over 20 iterations.

Fig. 6 shows the average time to rendezvous. We observe that the average TTR increases with c because the sequences get longer and the probability to reach the rendezvous quickly is progressively lowered. The TTR for the AMRCC with variable slots is always significantly lower than the one of the sequence-based rendezvous, with a peak improvement of 68% of the enhanced AMRCC with variable slots over the sequence-based rendezvous. Note that the performance does not degrade quickly with the number of channels n, mainly seen due to the improvement in the AMRCC with variable slots. By considering the duration of the highest ranking channel equal to the number of channels n, the longer sequences for high values of n are compensed by a longer permanence of a node over

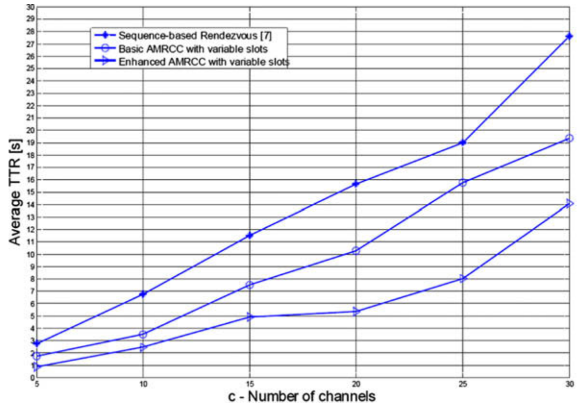

Fig. 6. Average time to rendezvous.

the highest ranking channel. The TTR depends on Δ_h, which is the duration of one hop, in this case equal to 5 s. The average TTR is proportional to the Δ_h.

Fig. 7 shows the average number of successful CR user transmissions versus the mean waiting time for the CR user Poisson traffic λ_{CR}. The parameters have been set as follows: $\Delta_{ob} = 15,000$ s, $\Delta_h = 5$ s, $\Delta_s = 15$ s, $T_{out,s} = 500$ s, λ_{PU} comprised in the interval [5; 20] s, λ_{CR} spanning in the interval [5; 15] s, $n = 10$, $off_{max} = 5$ s and $c = 5$. All the solutions show a decreasing curve with the increasing of the λ_{CR} as we expected and, especially for low values of λ_{CR}, i.e. for greedy traffic, both the basic and the enhanced AMRCCs with variable slots show higher values compared to the sequence-based rendezvous. The peak improvement of the enhanced AMRCC with variable slots over the sequence-based rendezvous is of almost 20%.

Fig. 8 shows the average number of CR user successful transmissions versus the average waiting time for PU Poisson traffic λ_{PU}, i.e. CR user activity versus PU activity. The parameters have been set as follows: $\Delta_{ob} = 15,000$ s, $\Delta_h = 5$ s, $\Delta_s = 15$ s, $T_{out,s} = 500$ s, λ_{PU} comprised in the interval [5; 75] s, $\lambda_{CR} = 5$ s, $n = 10$, $off_{max} = 5$ s and $c = 5$. Obviously, when λ_{PU} increases, i.e. the PU traffic decreases, the number of CR user successful transmissions increases signif-

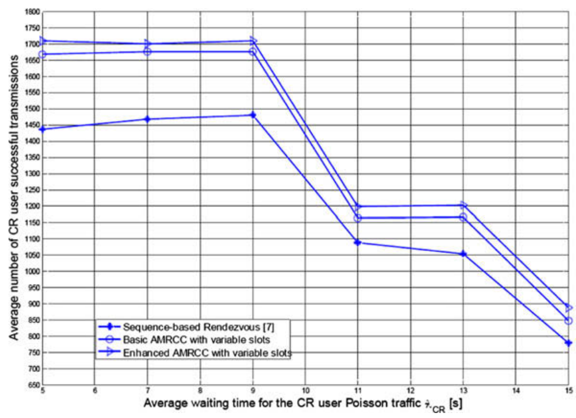

Fig. 7. Average number of successful CR user transmissions vs. the average waiting value for CR user traffic.

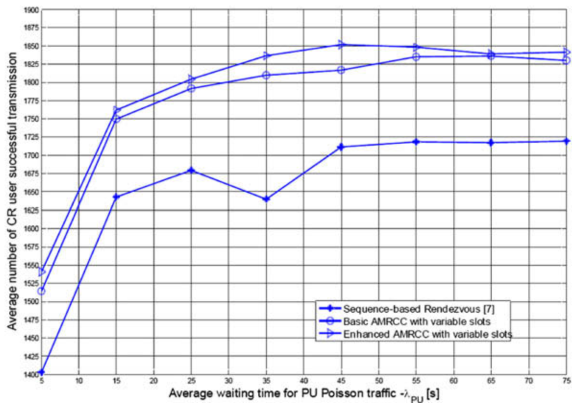

Fig. 8. Average number of successful CR user transmissions vs. the average waiting value for PU traffic.

icantly. For all the values of c, both the basic and enhanced AMRCC with variable slots outperform the sequence-based rendezvous. Our enhanced AMRCC solution with variable slots achieves a peak improvement of 13% over the sequence-based rendezvous.

Note that, even if the enhanced AMRCC with variable slots outperforms the basic AMRCC with variable slots, the difference between the curves is not significant. The reason is that, although the enhanced solution has lower average TTR and thus more CR user transmissions are attempted, the increased number of collisions with the PU packets leads to a similar performance in the results.

Figs. 9–11 show snapshots of the protocol behavior in the interval of observation $[0; 500]$ s. The above figures show how the CR user activity reacts to different PU traffic loads. In all the snapshots, the average waiting time for CR user Poisson traffic $\lambda_{CR} = 5s$, while different loads are considered for the PU traffic.

In particular, Fig. 9 refers to a high PU traffic load, where $\lambda_{PU} = 5$ s, Fig. 10 refers to a medium PU traffic load, with $\lambda_{PU} = 30$ s, and Fig. 11 refers to a low PU traffic load, with $\lambda_{PU} = 70$ s. We observe that when the PU activity decreases, the number of successful CR user transmissions increases, and the number of collisions between CR user and PU packets significantly decreases.

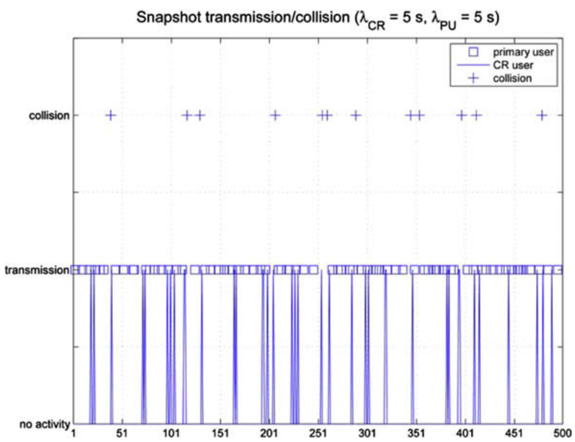

Fig. 9. CR user transmissions/collisions in case of high PU activity.

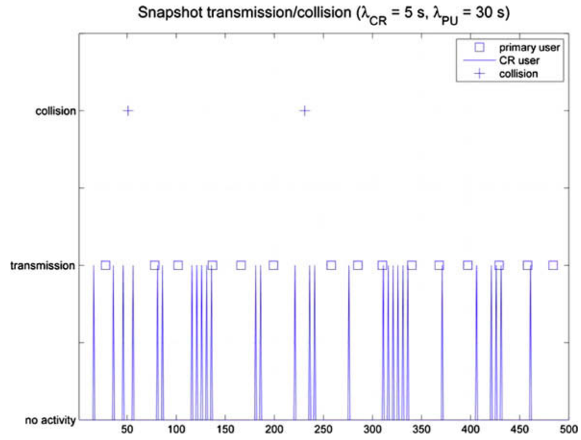

Fig. 10. CR user transmissions/collisions in case of medium PU activity.

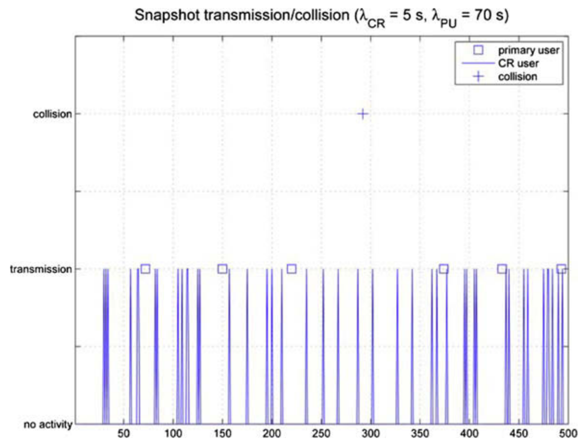

Fig. 11. CR user transmissions/collisions in case of low PU activity.

6. Conclusions and future directions

In this paper we proposed the AMRCC, a new adaptive multiple rendezvous control channel aiming at the reduction of the time to rendezvous and the increase of the overall network performance compared to the classical MRCCs. Our hopping sequence formation was adaptive based on the sensing results, where the channels with less PU activity occur more often. Two different approaches were considered: the basic AMRCC and the enhanced AMRCC, and both have the advantages not to require synchronization, are scalable and robust to dynamic PU activity. Also a slot duration optimization of the basic and enhance AMRCC has been proposed which avoids a significant performance degradation when the number of available channels gets high, by making the duration of the slots variable. Overall, the design of *always available* CCC is an important pre-requisitie for higher layer protocols in the area of CR networks.

Acknowledgements

The authors would like to thank Professor Ian F. Akyildiz for his advice and support.

References

[1] I.F. Akyildiz, W.-Y. Lee, K.R. Chowdhury, CRAHN: cognitive radio ad hoc networks, Ad Hoc Networks (Elsevier) Journal (July) (2009) 810–836.
[2] T. Chen, H. Zhang, G.M. Maggio, I. Chlamtac, CogMesh: A cluster-based cognitive radio network, in: Proc. of IEEE DySPAN, April 2007, pp. 168–178.
[3] M.C.-H. Chek, Y.-K. Kwok, On adaptive frequency hopping to combat coexistence interference between Bluetooth and IEEE 802.11b with practical resource constraints, in: Proc. of 7th International Symposium on Parallel Architectures, Algorithms and Networks, May, 2004, pp. 391–396.
[4] T. Chen, H. Zhang, M.D. Katz, Z. Zhou, Swarm intelligence based dynamic control channel assignment in cogmesh, in: Proc. of IEEE International Conference on Communications Workshops (ICC Workshops), May 2008, pp. 123–128.
[5] C. Cordeiro, K. Challapali, C-MAC: a cognitive MAC protocol for multichannel wireless networks, in: Proc. of IEEE DySPAN, April 2007, pp. 147–157.
[6] C. Cormio, K.R. Chowdhury, A survey on MAC protocols in cognitive radio wireless networks, Ad hoc Networks (Spring) (2009).
[7] L.A. DaSilva, I. Guerriero, Sequence-based rendezvous for dynamic spectrum access, in: Proc. of the 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN'08), October 2008, pp. 1–7.
[8] Y.R. Kondareddy, P. Agrawal, Synchronized MAC protocol for multi-hop cognitive radio networks, in: Proc. of IEEE International Conference on Communications (ICC), May 2008, pp. 3198–3202.
[9] L. Ma, X. Han, C.-C. Shen, Dynamic open spectrum sharing for wireless ad hoc networks, in: Proc. of IEEE DySPAN, November 2005, pp. 203–213.
[10] L. Ma, C.-C. Shen, B. Ryu, Single-radio adaptive channel algorithm for spectrum agile wireless ad hoc networks, in: Proc. of IEEE DySPAN, April 2007, pp. 547–558.
[11] B. Treister, A. Batra, K.C. Chen, O. Eliezer, Adaptive frequency hopping: a non-collaborative coexistence mechanism, in: IEEE P802.15 Working Group Contribution, IEEE P802.15-01/252r0, Orlando, FL, May 2001.

Claudia Cormio was born in Terlizzi (BA) on 16 January 1982. She received, with honours, the MS Degree in Electronic Engineering (Telecommunication field) from Politecnico di Bari (BA–Italy) in March 2006.
From October 2005 to February 2006 she collaborated with the Planetek Italia Company in Bari for the thesis titled "Tracking informations using UDP Protocol over GPRS Channel".
In May 2006, she joined the Telematics Lab, Politecnico di Bari, where she completed her PhD in 2009. She has also been a visiting researcher at the Broadband Wireless Networking Laboratory (BWN-Lab) at Georgia Institute of Technology.
Her current research interests include MAC protocols in Cognitive Radio Wireless Networks and reliable Common Control Channel management in Cognitive Radio Wireless ad hoc Networks.

Kaushik R. Chowdhury graduated with B.E. in Electronics Engineering with distinction from VJTI, Mumbai University, India, in 2003. He received his M.S. in Computer Science from the University of Cincinnati, OH, in 2006, and Ph.D. from the Georgia Institute of Technology, Atlanta, GA in 2009. His M.S. thesis was given the outstanding thesis award jointly by the ECE and CS departments. He has also won the BWN researcher of the year award during his Ph.D. in 2007, and the best paper award in the AHSN symposium in IEEE ICC 2009. His expertise and research interests lie in wireless cognitive radio ad hoc networks, resource allocation in wireless multimedia sensor networks, and bio-medical applications of sensors. He is a member of the IEEE.