Agnostic Broadcast Rendezvous for Cognitive Radio Networks Using Channel Hopping

Raphael M. Guedes and Marcel W. R. da Silva and Pedro S. Coutinho and José F. de Rezende
Universidade Federal do Rio de Janeiro (UFRJ)
P.O. Box 68.504 – Rio de Janeiro, RJ 21.945–970
Emails: {raphael,marcel,coutinho,rezende}@land.ufrj.br

***Abstract*—Cognitive radios (CRs) are a promising solution for the spectrum scarcity problem. These radios access the spectrum in an opportunistic manner in order to take advantage of frequency bands, i.e. channels, left unused by primary users while not interfering with them. This opportunistic access model requires CRs to switch among channels searching for new opportunities and making contact with their neighbors to exchange the information needed for spectrum management. This search process, known as rendezvous (RV), is termed blind RV when it does not use a common control channel (CCC) or a centralized solution. Several algorithms have been proposed to deal with the blind RV problem. However, most of them do not deal with broadcast transmissions, that is, when a CR wants to exchange information with all its neighboring nodes. This paper proposes a broadcast mechanism that operates on top of different one-to-one RV algorithms but being agnostic to the one in use, i.e., any such mechanism can be combined with it. Simulation results show that the proposed mechanism achieves a favorable performance while inheriting the good properties of the underlying one-to-one RV algorithm.**

I. Introduction

The use of cognitive radios (CRs) is currently considered a promising solution to the frequency spectrum scarcity problem. However, the opportunistic model of spectrum usage employed by the users of such devices brings great challenges to their designers. Usually, these devices must scan multiple channels in search of spectrum opportunities and then coordinate themselves in order to exchange information. Additionally, secondary users must primordially avoid causing interference to primary users. Because of this, a secondary user must immediately switch its operating channel if a primary user arrives in that channel. In scenarios where primary users make dynamic use of the spectrum and secondary users are equipped with only one radio interface, the latter must be constantly switching their operating channels seeking opportunities.

A commonly used channel switching technique, mostly employed in multiple channel ad hoc networks, is known as channel hopping (CH). In this technique, the radio repeatedly switches channels according to a certain sequence, which defines the *virtual channel* used by the device. This enables a better utilization of the multiple channels and when the CH sequences used by the various radios are chosen appropriately, it allows simultaneous transmissions in *virtual channels* that have orthogonal sequences. When employed by cognitive radios, this technique also permits secondary users to communicate even in situations where spectrum opportunities are not the same among all users, such as when secondary users are in the interference ranges of different primary users. This technique requires secondary users to be aware of each other in order to set up a communication link between them. For this, users who want to communicate must be in the same channel at the same point in time.

The encounter of two or more radios in a channel is called rendezvous (RV). In the case of cognitive radios, it is assumed that there is no centralized control entity or a dedicated common control channel for the exchange of information about each radio's hopping sequence. This makes the RV process harder, and it is known as blind rendezvous. Several studies in the literature deal with the blind RV problem [1]–[6], providing algorithms that generate channel hopping sequences that facilitate the rendezvous between two devices. The objective of these algorithms is to provide an upper bound for the time needed to perform the rendezvous. Some of these algorithms also try to better distribute the overlaps of the different channel hopping sequences in order to allow multiple pairs of devices to rendezvous simultaneously on different channels, reducing contention.

However, none of the cited algorithms deal directly with the problem of rendezvous between multiple devices, which would happen in a broadcast transmission. In the case of periodic broadcast transmissions, one solution would be to define a common channel hopping sequence among all the devices for this purpose. Then, the devices would temporarily switch to this sequence when they needed to perform a broadcast transmission. This alternative has several disadvantages related to the interval between periodic broadcasts. Very short intervals would result in a waste of bandwidth and longer intervals would result in higher latency to perform the broadcast. Another disadvantage is the contention that happens when different devices need to perform broadcast in the same channel sequence. The work in [7] uses this approach for broadcasting messages in a multi-hop and multi-user scenario.

For these reasons, we consider that on-demand broadcast transmissions are more convenient. In such case, one way to achieve this would be to repeatedly transmit the broadcast

This research was supported by FAPERJ, CAPES and CNPq.

978-1-4673-1564-7/12/$31.00 ©2012 IEEE

message performing RVs to every neighbor in a one-to-one fashion [5] in hopes of emulating a one-to-N RV. In this paper, we propose a mechanism for performing broadcast transmissions in channel hopping cognitive radio networks. This mechanism aims to facilitate broadcast transmissions using blind one-to-one RV mechanisms taking into account heterogeneous spectrum conditions among secondary users and dynamic spectrum usage carried on by primary users. The mechanism proposed in this paper has as its main characteristic the fact that it is RV mechanism agnostic, i.e., it can be used in combination with any blind one-to-one RV mechanism.

This paper is organized as follows. The next section presents related works. Section III describes the proposed mechanism. Section IV describes the model used in the performance evaluation and shows the numerical results obtained through simulation as well as a discussion about them. Finally, Section V presents the conclusions of this work.

II. Background and Related Works

In distributed Cognitive Radio Networks (CRNs), a control channel is required for coordinating spectrum management and channel contention between CRs. The work in [8] categorizes various control channel design schemes. In the *sequence-based control channel design* [8], control channels are allocated according to a random or predetermined channel hopping (CH) sequence. Thus, a control channel is dynamically established by multiple rendezvous among CRs. This design minimizes the impact of PU activity over the control channel due to the spectrum allocation's variation over time. Additionally, this approach allows the simultaneous communication of multiple CRs in the same neighborhood by allocating different CH sequences to each control channel. Numerous works in the literature tackle the rendezvous problem with this design scheme by using different strategies in the construction of the CH sequences [1]–[5], [7], [9]. The work in [3] identifies five different system models under which rendezvous can occur according to system capabilities, spectrum policies and environmental conditions.

Most of the existing RV algorithms assume a time-slotted system, in which at each time slot CRs hop among channels to attempt rendezvous with their potential neighbors. In a CH scheme where each CR hops among channels randomly [1], when two CRs hop to the same channel at the same timeslot, a rendezvous (RV) takes place in case the channel is free from PU presence. Otherwise, the nodes shall continue hopping until a RV occurs. During the RV, the nodes exchange control information.

RV schemes can be categorized as synchronous [4], [5] or asynchronous [2], [3], [7] whether they require that the CRs start their CH sequences at same time or not. In synchronous systems, coordination of CR users is required before rendezvous to achieve time-synchronization. Most RV systems use a static CH sequence [2], [4], [5]. However, some algorithms [3], [7] provide sequences that change over time in rounds.

The next subsections detail the four algorithms that have been used in this proposal as underlying one-to-one RV mechanisms. The two first algorithms are asynchronous and the remaining two are synchronous.

A. Modular Clock

The modular clock [3] is an algorithm that uses prime number modular arithmetic to make time-to-rendezvous (TTR) bound guarantees. The algorithm begins by selecting a channel randomly from the list of available channels. After each time slot, CR i hops "forward" r_i channels in the channel list wrapping around when it reaches a channel greater than a threshold. After $2 * p_i$ time slots, where p_i is the lowest prime greater than or equal to the number of channels, a new r_i value is randomly chosen in $[0, p_i)$.

B. Sequence-based Rendezvous (SeqR)

This scheme [2] builds the CH sequences in the following manner. It first selects a permutation of the set of channels, which has N elements. Then this permutation is repeated N times, interspersed with its own elements. For example, if the selected permutation when $N = 3$ is $\{2, 0, 1\}$, the initial sequence would be the first element of the permutation, then the whole permutation followed by the second element, followed by the permutation again, and so forth. So, the resulting sequence for this example would be $\{2, 2, 0, 1, 0, 2, 0, 1, 1, 2, 0, 1\}$. Every sequence generated by the SeqR scheme has its period equals to $N \times (N + 1)$ slots. Then each CR uses a different cyclic rotation of the initial CH sequence as its own sequence and a maximum of $N \times (N+1)$ different sequences can be generated. This scheme guarantees that one-to-one rendezvous will eventually occur, as long as at least one channel is free from primary users.

C. Quorum-based Channel Hopping

The work in [4] presents a scheme based on quorum systems to define CH sequences with the aim of increasing the number of RVs. A quorum is one element of a set S that satisfies the following intersection property: $p \cap q \neq \emptyset$, $\forall p, q \in S$. With this property, a cyclic quorum system can be constructed from a relaxed cyclic (n, k)-difference D which is contained in the set of channels, where n is the number of channels and k is the number of elements in D. For example, from the channel set $Z_3 = \{0, 1, 2\}$, $n = 3$ and $k = 3$ one can find the quorum set $S = \{\{0, 1\}, \{0, 2\}, \{1, 2\}\}$. Each element of S provides a sequence as shown in Figure 1. We have three sequences u, v and w that have overlapping channels, so each sequence can be assigned to a node guaranteeing that RVs will occur. In the figure, the blanks represent the cases where the channels are assigned randomly.

Multiple channel hopping sequences constructed from quorum systems can have greater overlapping to facilitate the rendezvous of two or more CR users with reduced and bounded average time-to-rendezvous (TTR).

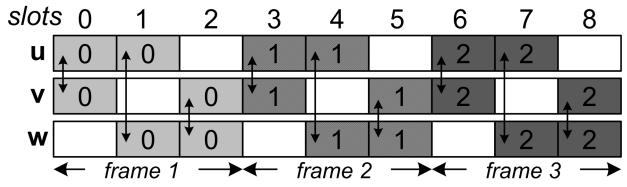


Fig. 1. Example of sequences from a quorum set.

D. Slotted Seeded Channel Hopping (SSCH)

In this algorithm [5], each node uses one or multiple (chan,seed)-pairs to determine its CH sequences, where *chan* is the initial channel and *seed* is the rate at which channel hops in the sequence. $N-1$ seeds are allowed, where N is the number of channels. Each sequence period includes a parity slot in which the used channel is determined by the seed value. Thus, all nodes with the same seed are guaranteed to rendezvous on this slot. For one (channel,seed)-pair, the period of the sequence is $N+1$ time slots and nodes rendezvous exactly once within a period when using different seeds.

III. Proposed Mechanism

The proposed RV mechanism is based on the retransmission of broadcast messages on all channels of a channel hopping sequence, where this sequence is chosen with the objective of minimizing the time-to-rendezvous (TTR) between the broadcasting node and all of its neighbors. This sequence is referred to as the *broadcast CH sequence*.

This mechanism assumes that all secondary users on the network use the same blind one-to-one RV mechanism in a role-based model [3]. In this RV model, it is considered that the nodes can assume different roles and in each role they can use different channel hopping sequences. In this work, the idle nodes, i.e. nodes that currently have nothing to transmit use a channel hopping sequence called *idle CH sequence*. When a node wishes to rendezvous or to transmit a broadcast message, it switches to an appropriate sequence, and then returns to the idle CH sequence after performing its task. This sequence is individual to each node and is determined by the blind one-to-one RV mechanism used.

We consider that every secondary user in the network performs periodic RVs with their neighbors and exchange information regarding their idle CH sequences and channel availability. Each node transmits the parameterization of the one-to-one RV mechanism used in a way that its idle CH sequence can be reproduced by its neighbors. It also informs its perspective of each channel's primary activity, more precisely the probability of that channel being free from primary users, p_{idle}, to its neighbors. This information permits the nodes to calculate the broadcast CH sequence according to Algorithms 1 and 2.

We propose two algorithms that generate a broadcast CH sequence from the exchanged idle CH sequence and channel availability information, namely Greedy Simple and Greedy p_{idle}. The purpose of both is to come up with a broadcast CH sequence that allows the broadcasting node to RV with the largest number possible of its neighbors. Both algorithms scan all neighbor's idle CH sequences and choose the channel to be used in each slot of the broadcast sequence as the channel that enables RV with the greater number of nodes in a greedy manner. At each iteration, when a node has already been reached, its idle CH sequence is no longer taken into account, and this goes on until all nodes are reached. If all nodes are reached before the broadcast CH sequence has the same size as the idle CH sequences, the algorithm tries once again to reach all nodes until the CH sequence achieves the desired size, when the algorithm stops.

The case where there is a tie when choosing the channel to be used in the next slot of the broadcast sequence, i.e. there

Algorithm 1: Greedy Simple broadcast sequence selection.

```
1  input: RADIOS, broadcastingnode, SEQSIZE,
   CHANNELS, SEQ(channel, slot)
2  init sequence = { }
3  init reachedNodes = ∅
4  while (|sequence| < SEQSIZE) do
5      select channel C where broadcastingnode reaches
       greater number of neighbors ∉ reachedNodes
6      append C to sequence
7      add all reached neighbors to reachedNodes
8      if (reachedNodes == RADIOS) then
9          reachedNodes = ∅
10 return sequence
```

Algorithm 2: Greedy p_{idle} broadcast sequence selection.

```
1  input: RADIOS, broadcastingnode, SEQSIZE,
   CHANNELS, SEQ(channel, slot),
   P_idle(node, channel)
2  init sequence = { }
3  init reachedNodes = ∅
4  for (node in RADIOS) do
5      init cumP_idle(node) = 0.0
6  while (|sequence| < SEQSIZE) do
7      select channel C where broadcastingnode reaches
       greater EXPECTED number of neighbors
       ∉ reachedNodes
8      for (node in neighbors reached) do
9          cumP_idle(node) =
           cumP_idle(node) + P_idle(node, C)
10         if (cumP_idle(node) >= 1.0) then
11             add node to reachedNodes
12     append C to sequence
13     if (reachedNodes == RADIOS) then
14         reachedNodes = ∅
15 return sequence
```

are two or more channels which allow reaching the maximum number of nodes is when the algorithms begin to differ. The Greedy Simple algorithm simply chooses the channel with the lowest identification number (ID), whereas the Greedy p_{idle} algorithm uses each neighbor's estimate of the idle rate of the channels to select the channel which is most likely to be idle for most of the nodes.

Moreover, in the Greedy Simple algorithm a node is considered as reached when its idle CH sequence has one overlapping slot with the broadcast sequence, that is, both sequences have the same channel in the same slot position. On the other hand, the Greedy p_{idle} algorithm only considers a neighbor as reached and stops taking it into account when choosing the broadcast sequence when the sum of the p_{idle}'s of that node for that slot exceeds 1.0. This means it only considers a node as reached when there is a high likelihood of being in the same channel in the same slot while that channel is idle from PU activity. When a secondary node uses a broadcast CH sequence to transmit a message, it only transmits on a given channel at a given time slot if that channel is free from PU activity in that slot. Otherwise, it waits for the next time slot, repeating this procedure until it has gone through the whole broadcast CH sequence. However, because of PU activity, the broadcast message may not reach all nodes if the sequence is followed only once, and the broadcast CH sequence may be used repeatedly in an effort to increase the odds of reaching all neighbors.

IV. Simulation Model and Results

The implemented simulation model emulates the behavior of a node sending periodic broadcast messages to its neighbors according to a role-based RV model. For that reason, the simulation scenario consists of $X + 1$ secondary users, with one node (node B) transmitting messages in a broadcast manner and its X neighbors acting as receiver nodes. In each simulation, the X receiver nodes utilize individual idle CH sequences given by one of the following blind one-to-one RV mechanism: modular clock (`mclock`), sequence-based rendezvous (`seqr`), quorum-based channel hopping (`quorum`), or slotted seeded channel hopping (`ssch`). Each of the X receiver nodes has its own idle CH sequence, which is obtained by selecting individual initial parameterization of the RV mechanism in use on that simulation.

We assume that in past rendezvous node B has already exchanged information about its neighbors idle CH sequence parameterization and their perspective of each channel availability. We simulated two cases, where node B uses or neglects that privileged information when it assumes the role of transmitting broadcast messages. By using this privileged information, node B can operate in two different modes, by calculating a broadcast CH sequence using one of the two algorithms proposed in Section III: greedy simple (`simple`) or greedy p_{idle} (`pIdle`). Otherwise, in the case where this previous knowledge is neglected, node B uses a CH sequence for broadcast given by the same one-to-one RV mechanism that was used by its neighbors to determine their idle CH sequences. It is the best node B can do without any knowledge from its neighbors CH sequences parameterization, since this approach guarantees at least the good properties of the one-to-one RV mechanism in use.

All nodes in the simulation look for dynamic spectrum opportunities in N channels. Each channel is utilized by a primary user (or primary network) whose behavior follows an `ON-OFF` model. The duration of each state is exponentially distributed with mean μ_{ON} and μ_{OFF}, respectively. Thus, the probability of channel i being idle is given by $p_{idle} = \frac{\mu_{OFF,i}}{\mu_{ON,i}+\mu_{OFF,i}}$. These idle probabilities are randomly generated for each channel and are uniformly distributed between 0.0 and 1.0.

In order to determine the PU influence on the secondary communication, the following model was used. Node B is initially positioned in the center of a circle of radius R which represents its communication range. Its X neighbors are then uniformly positioned within that circle while keeping a minimum distance of D_{min} between each other. Then N primary nodes, one for each channel, are disposed randomly within a square area concentric to the circle. The minimum distance D_{min} is also kept between them. All secondary nodes within range R of a given primary node i will sense channel i with its corresponding idle probability and the remaining nodes will sense that channel with $p_{idle} = 1.0$. We consider that every secondary node is able to estimate the PUs idle rate of each channel from periodic sensing operations on such channels as in [10].

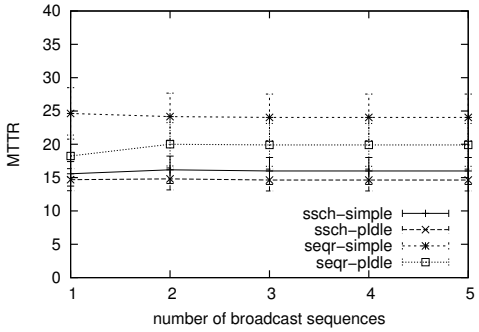
Because of the dynamic influence of the primary users, there is no guarantee that by the end of one broadcast CH sequence all neighbors received the broadcast message. Therefore, a possible solution would be to extend the time node B stays in the broadcast role. To evaluate this idea, we introduced a new parameter in the simulations called *number of broadcast sequences*, which indicates how many times node B will repeat the broadcast CH sequence, staying in the broadcast role.

An important metric for evaluating CH schemes, which is measured in simulations, is the *maximum time-to-rendezvous* (MTTR). It consists of the maximum time in number of time slots (or channels in the sequence) needed for two nodes to achieve rendezvous. In the broadcast case, the MTTR is defined as the maximum time the broadcasting node needs to rendezvous with all of its neighboring nodes in a simulation run. Besides MTTR, we also evaluate in simulations the *average time-to-rendezvous* (ATTR), which analogously to the MTTR, is defined as the average time needed to rendezvous with all neighboring nodes.

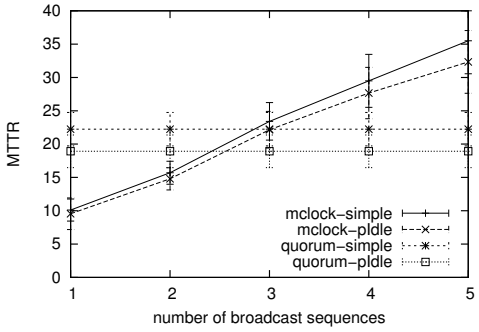
Since we assume that broadcast is not completely reliable, i.e. some broadcasts do not reach all neighboring nodes, we also introduce a new metric for evaluating broadcast RVs, named success rate. This metric provides the percentage of broadcast RVs in which all neighbors were encountered. Therefore, the MTTR is only computed on the successful broadcast RVs.

The simulation results presented in this Section were obtained in scenarios with: $N = 13$ channels, $\mu_{OFF} = 5$

slots, $R = 250$ meters, and $D_{min} = 20$ meters. We varied the number of broadcast sequences from 1 to 5, and varied X (number of node B's neighbors) from 2 to 8. In each simulation run, node B sends 1000 broadcast messages, i.e. node B assumes the broadcast role 1000 times. All the results presented in graphs are the average of 100 simulation runs with 95% confidence intervals.



(a) SSCH and SeqR

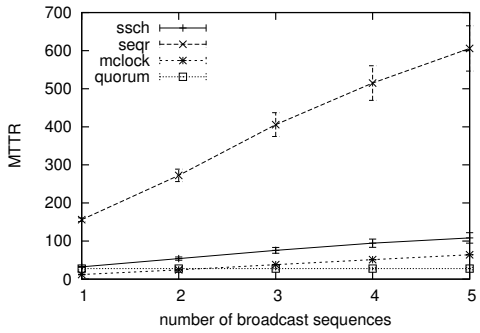


(b) Quorum and Modular Clock

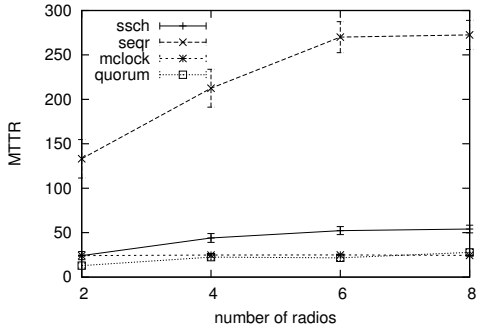
Fig. 2. MTTR for the two algorithms - 8 CR nodes

Figures 2(a) and 2(b) show the performance of both proposed algorithms, `pIdle` and `simple` for all one-to-one RV schemes, in terms of maximum time-to-rendezvous (MTTR) to reach all neighbors. It can be seen that `pIdle` outperforms `simple` since it takes into account the PUs activity. This difference is observed in all of the obtained results. For this reason, we will use only the results obtained with the `pIdle` algorithm to evaluate the performance of this work's proposal. However, it can be noticed that the performance of `simple` is close to the one obtained by the other algorithm and it has the advantage of not requiring the estimate of the channel occupation from the perspective of each neighboring node.

Figure 3 presents the MTTR when the RV schemes are used without our broadcast CH sequence proposal. It can be noted that the `seqr` scheme has the greatest MTTR values, even when only one broadcast sequence cycle is used. Its poor performance in this aspect is due to the fact that this scheme generates CH sequences with very high number of slots. The remaining schemes present MTTR values lower than 100 slots for all the parameters used. In Figure 6(b), we see that the



(a) 8 CR nodes



(b) 2 broadcast sequences

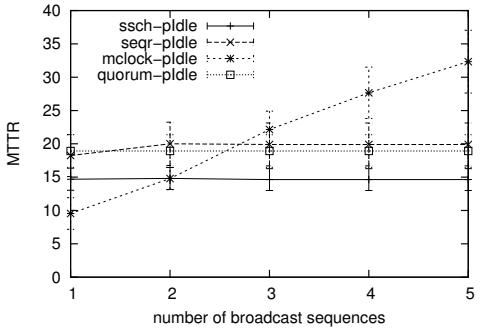
Fig. 3. MTTR without the proposal

MTTR of the `mclock` scheme gets worse as the number of broadcast sequences used increases. This happens because its broadcast success rate increases, while at the same time higher MTTRs are included in the results, which only take into account the successful broadcast cases.

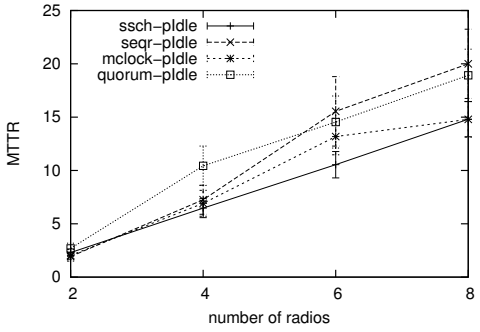
Figure 4 presents the MTTR results when using our broadcast CH sequence proposal. It can be observed that the performances of all RV schemes are enhanced by the use of the proposed broadcast sequence algorithm. It can be noted in Figure 4(a) that the `mclock` scheme once again has worse MTTRs as the number of cycles of the broadcast sequence increases. This occurs because increasing this number of cycles makes the broadcast success rate of this mechanism much higher, as it can be seen in Figure 10, and this leads to the inclusion of higher MTTRs in the results, which only compute successful broadcast cases.

Figure 5 shows the MTTR reduction percentage achieved by making use of our broadcast CH sequence proposal combined with each different RV scheme. It can be seen that the `seqr` scheme, which has the worst absolute performance when compared to the others, is the one that has its performance increased the most when coupled with our proposal. For all RV schemes, the smallest percentage of reduction is about 30%, which shows the importance of the performance improvement offered by using our proposal.

Figure 6 presents the average time-to-rendezvous (ATTR) when the RV schemes are used without our broadcast CH sequence proposal. The behavior shown is similar to that

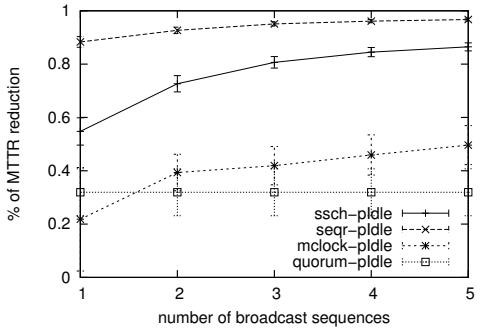


(a) 8 CR nodes

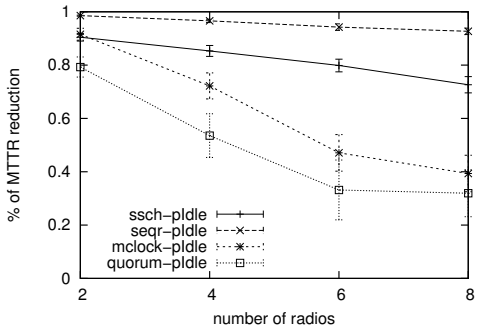


(b) 2 broadcast sequences

Fig. 4. MTTR with the proposal

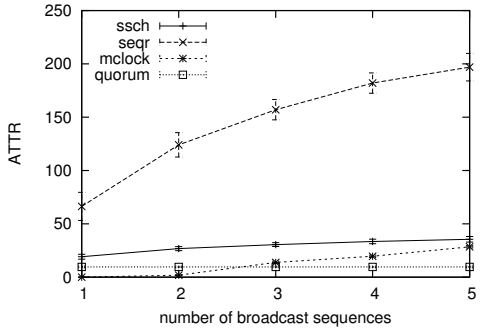


(a) 8 CR nodes

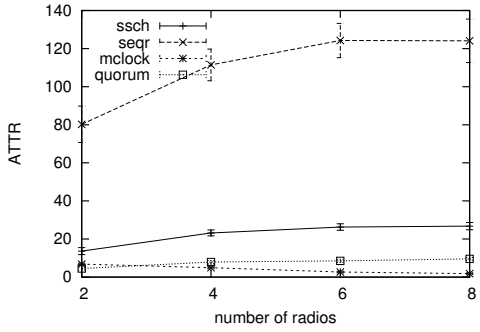


(b) 2 broadcast sequences

Fig. 5. MTTR reduction



(a) 8 CR nodes



(b) 2 broadcast sequences

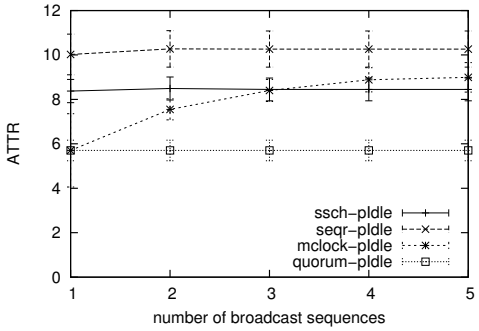
Fig. 6. ATTR without the proposal

of Figure 3, where the `seqr` scheme presents the worst performance, while the best performances are achieved by the `quorum` and `mclock` schemes. In Figure 6(b), it can be seen that the `mclock` scheme's ATTR also gets worse as the number of broadcast sequences used increases. This happens because its broadcast success rate increases, while at the same time higher ATTR values are included in the computations.

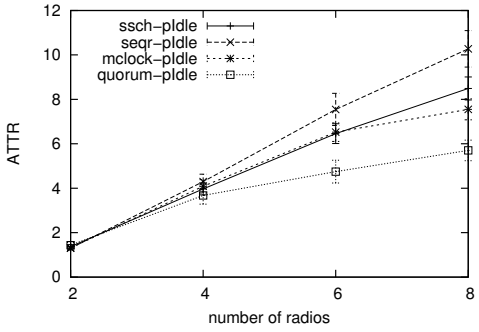
Figure 7 shows the ATTR results when using our broadcast CH sequence proposal. It can be seen that the performances of all RV schemes benefit from the use of the proposed broadcast sequence algorithm. In Figure 7(a), we can see that the `mclock` scheme has higher ATTRs as the number of cycles of the broadcast sequence increases. The happens because increasing this number of cycles makes the broadcast success rate of this mechanism become higher, as shown in Figure 10, and this causes the inclusion of higher ATTRs in the results, which only consider successful broadcast cases.

Figure 8 shows the ATTR reduction percentage caused by using our broadcast CH sequence proposal coupled with each different RV scheme. We see that the `seqr` scheme, which has the worst absolute performance when compared to the others, is the one that benefits the most from being combined with our proposal. For all RV schemes, the smallest reduction percentage is about 50%, which shows that the performance improvement offered by using our proposal is very significant.

Figure 9 shows the broadcast success rate of the different RV schemes without our `pIdle` proposal. The scheme that presents the worst delivery rate is `mclock`, with its best value

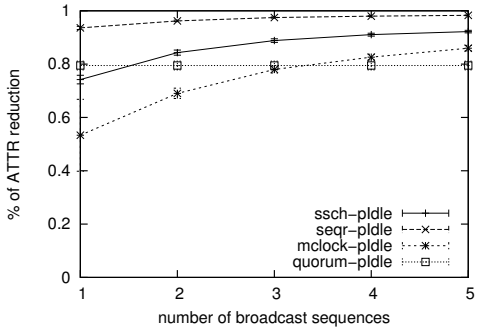


(a) 8 CR nodes

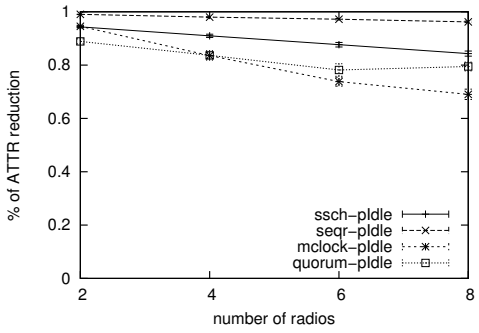


(b) 2 broadcast sequences

Fig. 7. ATTR with the proposal

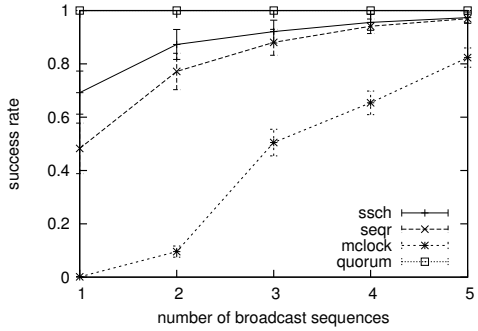


(a) 8 CR nodes

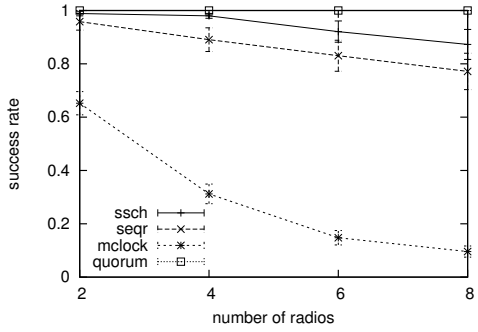


(b) 2 broadcast sequences

Fig. 8. ATTR reduction



(a) 8 CR nodes



(b) 2 broadcast sequences

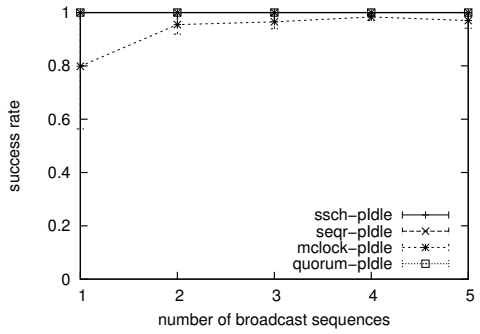
Fig. 9. success rate without the proposal

around 80%. The `quorum` scheme has 100% delivery rate even without using our broadcast CH sequence algorithm. This is due to the fact that this scheme has a very high degree of overlapping between each node's CH sequence. All schemes achieve better success rates when increasing the number of broadcast sequence cycles used. In Figure 9(b), we observe that the success rate gets worse as the number of radios increases, because it gets harder to deliver the broadcast message to all of the nodes. Once again the greater CH sequence overlapping of the `quorum` scheme allows it to maintain a high delivery rate, even with more radios in the secondary network.

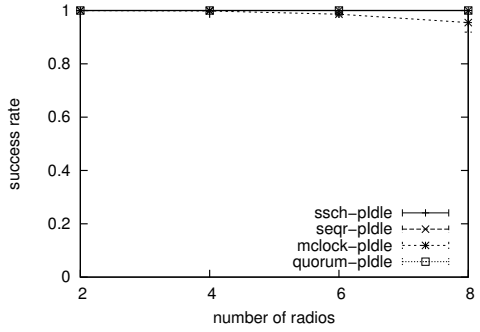
In Figure 10, we have the broadcast success rate of the different RV schemes when used in combination with our `pIdle` proposal. With the exception of the `quorum` scheme, which already had 100% success rate without our proposal, all RV schemes have improved considerably. The `ssch` and `seqr` schemes attained 100% delivery rates with our proposal and even the `mclock` scheme, which could not increase its success rate to 100%, has shown significant improvements.

V. Concluding Remarks

Broadcast in rendezvous systems using channel hopping (CH) is not a trivial problem. When this broadcast communication involves control messages dissemination in cognitive radio networks, the problem is even harder due to PU activity. One solution to this problem would be to have all CR nodes share the same CH sequence. However, this approach has many



(a) 8 CR nodes



(b) 2 broadcast sequences

Fig. 10. success rate with the proposal

disadvantages associated to the channel contention involved. Therefore, the straightforward alternative solution when CR nodes have different CH sequences is to have the broadcasting node retransmit the messages in each time slot of its own CH sequence so that they reach all of the neighboring nodes. However, when the sender CH sequence does not have a good overlap with its neighbors' sequences and PU activity is not taken into account, this strategy can provide a very loose bound on the time-to-rendezvous and hence add a lot of overhead.

In this paper, we propose two algorithms for constructing the broadcast CH sequence that take into account the number of overlapping channels and PU activity in each channel. These algorithms have the great advantage of being both simple and agnostic to the underlying one-to-one rendezvous scheme used. The algorithms were evaluated in terms of the achieved maximum and average time-to-rendezvous reduction when compared to the case in which the broadcasting node uses its own unmodified CH sequence to perform the broadcast. Simulation results show that both algorithms provide performance gains in terms of these metrics and also in terms of the number of reached neighbors, i.e. the success rate, when a limited number of broadcast sequences are employed.

REFERENCES

[1] M. D. Silvius, F. Ge, A. Young, A. B. MacKenzie, and C. W. Bostian, "Smart radio: Spectrum access for first responders," in *Proceedings of SPIE Defense and Security Conference*, Mar. 2008.

[2] L. A. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," in *Proceedings of IEEE DySPAN*, Oct. 2008, pp. 1 –7.

[3] N. C. Theis, R. W. Thomas, and L. A. DaSilva, "Rendezvous for cognitive radios," *IEEE Transactions on Mobile Computing,*, vol. 10, no. 2, pp. 216–227, Feb. 2011.

[4] K. Bian, J.-M. Park, and R. Chen, "Control channel establishment in cognitive radio networks using channel hopping," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 689–703, 2011.

[5] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *Proceedings of ACM MobiCom*. New York, NY, USA: ACM, 2004, pp. 216–230.

[6] Y. Zhang, Q. Li, G. Yu, and B. Wang, "ETCH: Efficient channel hopping for communication rendezvous in dynamic spectrum access networks," in *Proceedings of IEEE INFOCOM*, april 2011, pp. 2471 –2479.

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

[8] B. F. Lo, "A survey of common control channel design in cognitive radio networks," *Elsevier Physical Communication*, vol. 4, pp. 26–39, 2011.

[9] C. Cormio and K. R. Chowdhury, "Common control channel design for cognitive radio wireless ad hoc networks using adaptive frequency hopping," *Ad Hoc Networks*, vol. 8, no. 4, pp. 430–438, 2010.

[10] H. Kim and K. G. Shin, "Efficient discovery of spectrum opportunities with mac-layer sensing in cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 533–545, May 2008.