ICST Transactions on Mobile Communications and Applications Research Article ICST.ORG

Quorum system and random based asynchronous rendezvous protocol for cognitive radio ad hoc networks★

Sylwia Romaszko[1,*], Daniel Denkovski[2], Valentina Pavlovska[2], Liljana Gavrilovska[2]

[1]Institute for Networked Systems, RWTH Aachen University, Kackertstrasse 9, 52072 Aachen, Germany
[2]Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University in Skopje, Macedonia

Abstract

This paper proposes a rendezvous protocol for cognitive radio ad hoc networks, RAC^{2}E-gQS, which utilizes (1) the asynchronous and randomness properties of the RAC^{2}E protocol, and (2) channel mapping protocol, based on a grid Quorum System (gQS), and taking into account channel heterogeneity and asymmetric channel views. We show that the combination of the RAC^{2}E protocol with the grid-quorum based channel mapping can yield a powerful RAC^{2}E-gQS rendezvous protocol for asynchronous operation in a distributed environment assuring a rapid rendezvous between the cognitive radio nodes having available both symmetric and asymmetric channel views. We also propose an enhancement of the protocol, which uses a torus QS for a slot allocation, dealing with the worst case scenario, a large number of channels with opposite ranking lists.

Keywords: asymmetric channel view, asynchronous, cognitive radio ad hoc networks, quorum systems, rendezvous

Received on 15 March 2013; accepted on 14 November 2013; published on 16 December 2013

doi:10.4108/mca.1.3.e4

1. Introduction

A common control channel (CCC) in multichannel Cognitive Radio Networks (CRNs) supports the transmission coordination exchange and cooperation between the active CR users. It is aimed to facilitate neighbor discovery, e.g., control signaling, exchange of local measurements, channel sensing etc. However, such CCC existence in CRNs may not be always feasible. When using the CCC notion, a channel needs to be found that is accessible by the majority of CR nodes and it is not interrupted over a long period of time. However, these tasks are not feasible in a CR environment without any imposed assumptions, since CR nodes can have a different view of the channels occupied by incumbents and/or other secondary users. Another issue that arises when all nodes have chosen the same channel is the possibility of single channel bottleneck as well as the single point of failure.

Moreover, in Cognitive Radio Ad Hoc Networks (CRANs), the dynamic network topology, distributed multi-hop architecture, and time and location varying spectrum availability are the key factors [2]. Each Cognitive Radio (CR) user has a different spectrum availability according to the incumbent (Primary User, PU) activity, and it determines its actions based on its local observation. Therefore, rendezvous (RDV), the ability of two or more nodes to meet each other and establish a link, is a challenging task in CRANs.

This paper proposes a rendezvous protocol for CRANs that, firstly, utilizes the asynchronous and randomness properties of the RAC^{2}E protocol [3]. Secondly, it investigates the suitability of different channel search orders that are based on:

(i) random selection utilizing weights and utility functions,

★Part of this work was presented at ICST Conference, ADHOCNETS 2012, Paris, France, October 2012 [1] published in Ad Hoc Networks, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering Volume 111, 2013, pp 135-148.

*Corresponding author. Email: sar@inets.rwth-aachen.de

(ii) grid quorum-based channel mapping (gQS-RDV) protocol [4, 5] taking into account channel heterogeneity (in terms of channel quality),

(iii) torus QS optimization of the gQS-RDV slot allocation method.

Finally, it gives an insight of the performance of the proposed protocol in terms of the time-to-rdv (TTR)[1] and the probability of RDV for the different channel search orders, as well as the inter-rendezvous time variance. Moreover, we evaluate the protocol in a symmetric channel view (homogeneous channel availability) and asymmetric channel view (heterogeneous channel availability, i.e., having different number of idle/unoccupied channels per CR) cases.

The paper is organized as following. Section 2 presents the related work. In Section 3 we describe a Quorum System (QS) concept and relevant properties. Section 4 presents the RAC^{2}E-gQS protocol, its phases and optimization of the channel mapping phase. In Section 5 we evaluate the proposed protocol with and without the proposed optimization. The last section concludes this study.

2. Related work

In [6] one can find a comprehensive guidance on the application of quorum systems in wireless communications, and rendezvous issues in decentralized CRNs. In general, rendezvous approaches can be divided into three branches, first, non-quorum based solutions representing blind or pseudo-random RDV techniques ([7–9] and more sophisticated [10, 11]). To the second branch belong protocols proposed for a multi-channel Medium Access Control (MAC) handling multi -rendezvous [12] (i.e., multiple transmissions pairs can accomplish handshaking simultaneously), missing receiver problem [13] or medium allocation in a hostile and jamming environment [14]. All these protocols are based on cyclic quorum systems. A cyclic QS is proposed in [15] and it is based on the cyclic block design and cyclic difference sets in combinatorial theory [16]. The last branch consists of either quorum-based, or difference set-based, or Latin Square-based protocols proposed for CRNs ([17–21]). However, an asynchronous channel view (ACHv) is not explicitly handled in quorum-based schemes[2] and the *channel heterogeneity* is not considered in any related work approaches to the best of our knowledge.

The asynchronous operation of the cognitive radio networks and its effect on the rendezvous phase is not well investigated subject. The synchronization establishment in cognitive radio networks, especially in the distributed case, is a time and power consuming task, and therefore, the assumption that the CR nodes operate in time synchronized manner is not always justified. There are only a few papers considering the asynchronism of the CR nodes during the rendezvous phase. Most of them use probabilistic models to generate the channel hopping sequences of the operating CR devices. In the modular clock algorithm (MC) [8] and its modified version MMC [8] each CR node picks a proper prime number P and randomly selects a rate r which is less than P. Based on the two parameters, the user generates its channel search sequence via pre-defined modulo operations. The channel rendezvous sequence (CRSEQ) algorithm [23] uses a method based on triangle numbers and modular operations to calculate the channel hopping order. The ring-walk (RW) algorithm [10] represents each channel as a vertex in a ring. The CR nodes sweep through the ring visiting the vertices (channels) with different velocities and the rendezvous is guaranteed since the nodes with lower velocities will sooner or later be caught by the ones with higher velocities. In [11] the authors propose a jump-stay rendezvous algorithm for blind rendezvous, using jump-pattern and stay-pattern channel hoping sequences in each round. The CR nodes continuously "jump" on available channels during the jump-pattern and "stay" on a specific channel during the stay-pattern. In [20] two systematic approaches (symmetric and asymmetric) for designing asynchronous channel hopping (ACH) protocols are proposed, which address the asynchronous rendezvous problem. An asymmetric ACH system that uses an array-based quorum system is introduced instead of utilizing Latin squares as done in the prior work [19]. This quorum-based approach generates a significantly greater number of CH sequences than the approach using Latin squares. In the recent study on the blind rendezvous for tactical networks [24], the performances of the MMC [8] and Random Channel Access (RA) [25] algorithms are compared on a testbed using USRP [26]. It is found out that added asynchronism can have a large beneficial effects reducing time to rendezvous.

Up to the best of the authors' knowledge there is only a couple of papers focusing on quorum based asynchronous rendezvous [6, 17, 20, 22]. However, these papers as well as the aforementioned work dealing with the asynchronism, do not handle the channel heterogeneity in the generation of channel hopping sequences and do not handle the details of asynchronous operation and rendezvous between the devices. On the contrary, the RAC^{2}E-gQS takes into consideration the *heterogeneity* in terms of the

[1] TTR is an amount of time, measured in slots, within which cognitive radios meet each other once they began hopping, or after the last rendezvous on a channel.

[2] Visiting unavailable channels is the most frequently used approaches while dealing with ACHv in the related work (e.g., [18, 22]).

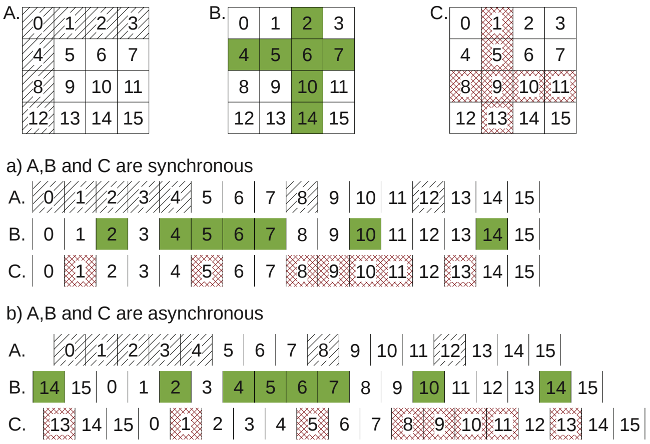

Figure 1. Example with grid-based quorums [4].

Table 1. 4x4 grid: Pair-On-Pair (PoP): quorum (0,0) in bold

0	5	11	15
4	1	7	13
10	6	2	9
14	12	8	3

Table 2. 4x4 grid: Diagonal (Diag): quorum (0,0) in bold

0	**4**	**8**	**12**
13	1	5	9
10	14	2	6
7	11	15	3

channel priority among the CR nodes. Furthermore, the proposed protocol covers the details of the operation of the nodes *prior the rendezvous* and *after the rendezvous*, i.e. the control channel operation.

3. Quorum systems

Quorum-based algorithms become popular, as the main asset of these algorithms is their resilience to node and network failures. The usual definition of a quorum system (QS) is given in [27]:

Definition 3.1. A *quorum system* Q under an universal set U, $U = \{0, 1, ..., n-1\}$ with n being a cycle length (frequently used Z_n symbol referring to $U = Z_n$), is a collection of non-empty subsets of U, called quorums, satisfying the *intersection property* $\forall A, B \in Q : A \cap B \neq \emptyset$.

There are different types of QSs, within which a *grid*-based QS proposed by Maekawa [28], is widely utilized in power-saving (PS) protocols. In this system, sites (elements) are logically organized in a grid in the shape of a *square*. A quorum for a requesting site contains the union of a row and a column that the requesting site corresponds to. For PS nodes we can divide their beacon intervals into groups, where each group includes n consecutive intervals and is organized in a $\sqrt{n} \times \sqrt{n}$ array in a row-major manner. Quorum intervals are picked along an arbitrary row and column from this array, where the remaining intervals are non-quorum intervals. Figure 1 depicts an example for 16 beacon intervals and three nodes, A, B, C, selecting different quorum intervals. If the clocks of the nodes are synchronized (case a), the intervals of the nodes overlap twice, e.g., A and B meet in 2 and 4 slots. If their clocks are not synchronized (case b) the A's slots still overlap with B's and C's slots.

A QS, which satisfies the *Rotation Closure Property (RCP)*, ensures that two asynchronous mobile nodes selecting any two quorums have at least one intersection in their quorums. The grid QS satisfies the RCP:

Definition 3.2. For a quorum R in a quorum system Q under an universal set $U = \{0, ..., N-1\}$ and $i \in 1, 2, ..., N-1$, there is defined:
$rotate(R, i) = (x + i) mod\ N | x \in R$. A quorum system Q has the *Rotation Closure Property* if and only if $\forall R', R \in Q, R' \cap rotate(R, i) \neq \emptyset$ for all $i \in 1, 2, ..., N-1$.

In [4] *Grid-Pair-on-Pair* (PoP) way of forming a grid was proposed as shown and Equation 1 and in Table 1.

$$f(x,y) = \begin{cases} x & \text{(x=y)} \\ f(x-1, y-1) + 2 & \text{(x=1..n-1, y=1..n-1 x!=y)} \\ \lfloor \frac{n+1}{2} \rfloor * 2 & \text{(x=1, y=0)} \\ (\lfloor \frac{n}{2} \rfloor * 2) + 1 & \text{(x=0, y=1)} \\ f(1,0) + ((x-1)n - \sum_{i=1}^{x-1} i) * 2 & \text{(x=2..n-1, y=0)} \\ f(0,1) + ((y-1)n - \sum_{i=1}^{y-1} i) * 2 & \text{(x=0,y=2..n-1)} \end{cases} \tag{1}$$

However, this grid does not satisfy the RCP. Hence, a node using this grid will encounter problems with RDV when there is no cycle alignment.
For example, while a node selects a quorum (0,0) as depicted in Table 1, a quorum $Q1 = \{0, 4, 5, 10, 11, 14, 15\}$ is chosen. Q1 does not satisfy the RCP, since for $i = 8$ there is no common element: $Q1 \cap rotate(Q1, i = 8) = \emptyset$, because of $\{0, 4, 5, 10, 11, 14, 15\} \cap \{8, 12, 13, 2, 3, 6, 7\} = \emptyset$. In all other cases there is at least one common element.

Therefore, *Grid-Diagonal* (Diag), shown in Table 2 and Equation 2, was designed in [4]. In this method the numbers are ordered according to the positive diagonal rule, i.e., elements are ordered according to

$$f(x, y) = ((y \times r) - ((r-1) \times x))\%(r \times r) \tag{2}$$

where $x = 0, ..., n-1$, $y = 0, ..., n-1$, and $r = \sqrt{n}$. This grid *does* guarantee the RCP. For instance for $Q2 = \{0, 4, 7, 8, 10, 12, 13\}$ from Table 2, there is always at least one common element, i.e., $Q2 \cap rotate(Q2, i) \neq \emptyset$ for all $i \in 1, 2, ..., 15$.

4. RAC^{2}E-gQS protocol description

In this section we describe RAC^{2}E-gQS composed of two main parts, first, a rendezvous-MAC protocol for asynchronous cognitive radios, and second being an optimization of the first part by utilization of a sequence -based channel mapping while handling a rendezvous phase. The former allows to benefit from the asynchronous and randomness properties, originally proposed in [3] as part of the RAC^{2}E protocol. The latter uses a grid Quorum System (gQS) properties, originally proposed in [4, 5], in order to deal with a rendezvous guarantee in a single cycle.

An original version of the RAC^{2}E protocol is proposed for nodes operating in an asynchronous and cooperative manner, i.e., a CR user utilizes information from its spectrum map to determine the best channel to be its control channel. Hence, if two CR nodes want to establish a direct link for a communication, they must exchange their spectrum maps first in order to select the mutually best channel (i.e. the channel having the lowest level of interference for both nodes).
In this work, we use only a part of the RAC^{2}E protocol, namely, asynchronous and randomness properties, while creating *asynchronous cycles* (and slots), and handling a control channel operation (*after the rendezvous* phase, thus, message exchange between nodes). Moreover, CRs can work in a non-cooperative manner, while allocating a channel. However, the reader should note that spectrum maps can be still utilized to determine the priorities of the channels and as a consequence the order of the channel in the channel ranking list. From now on, while talking about the RAC^{2}E, we refer to the aforementioned part of the RAC^{2}E-gQS protocol, and not original one designed in [3].

The mapping of channels into time slots, in the *prior the rendezvous* phase of the RAC^{2}E-gQS protocol, is a paramount task. This can be done using several methods considering the channel *priorities* based on the channel ranking lists created in the sensing phase by each node independently. And therefore, as the main contribution in this work we investigate the suitability of different channel search orders that are based on:

(i) random selection of channels using utility functions for channels priority- *UP*,

(ii) gQS based channel mapping strategies [4, 5] taking into account channel heterogeneity (in terms of channel quality).

(iii) torus QS optimization of the gQS slot allocation.

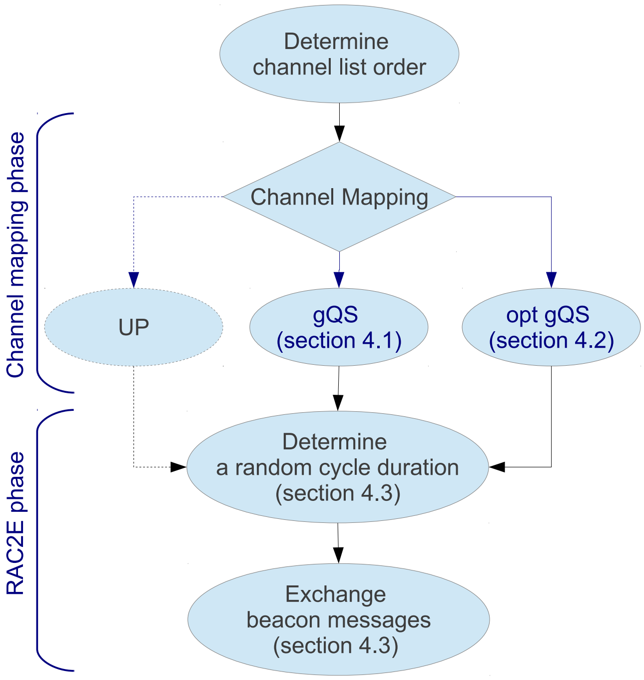

Figure 2. Asynchronous rendezvous-MAC protocol diagram with a choice of three different mapping methods, namely, UP, gQS, and optimized gQS.

The *UP* method randomly performs the channel mapping with probabilities guarantying that the channels, with respect to their priorities, will statistically be assigned to the same amount of slots per rendezvous cycle as the grid-quorum based methods. This channel mapping strategy is selected as a representative example since it probabilistically maps the channel to slots, oppositely on the regular grid mapping.

The combination of the RAC^{2}E protocol with the gQS mapping, RAC^{2}E-gQS, can yield a promising rendezvous protocol for asynchronous operation in a distributed environment assuring a rapid rendezvous between the CR nodes. However, in the worst case scenario, namely, a large number of of channels (~20 channels) and opposite channel raking lists, it works somewhat worse that the RAC^{2}E-UP combination, and therefore, we also proposed an optimization of RAC^{2}E-gQS in order to ensure its advantage in all case scenarios. This goal is accomplished by an enhancement of the channel mapping algorithm by utilizing a *torus* QS properties in a grid array. Moreover, note that the RAC^{2}E, proposed in [3], has already been implemented on a testbed platform (the results are presented in [3]), and the proposed protocol in this work, optimized RAC^{2}E-gQS, is under implementation and evaluation phase.

Figure 2 shows the steps of the RAC^{2}E-gQS protocol. The UP method is also visible in the diagram (with corresponding dashed arrows), however, since it is disadvantageous in all cases, normally it will probably not be a choice of a mapping method, but we plot it in the diagram for the sake of clarity which algorithms are analyzed in this work. Later in this work, we recommend the case scenarios in which either gQS or the optimized method should be used.

In the next subsections we describe the underlying algorithms used in the proposed protocol and its optimization.

4.1. Channel mapping phase

In this section we elaborate on a distributed grid QS based channel mapping algorithm. The outcome of the algorithm provides an input to the channel hopping sequences called channels-to-slots maps (or channel maps for the sake of simplicity). Each CR maps its channels according to the channel quality without any exchange of information, where the best channels get a higher priority. The best channel is mapped according to the chosen *quorum*. Hence, CRs that allocate a common best channel, while having the same number of available channels, will always meet as a result of the quorum intersection property (if satisfying the RCP they also always meet regardless cycle/slot misalignment). The period (cycle) N of a channel map depends on the number of channels r and it equals r^2 (selected from a $r \times r$ grid). Two channels-to-slots mapping methods are designed for three or larger number of channels (i.e. $r \geq 3$). In both methods, channels are mapped to grid indexes (Channel 1 (C1) is mapped to index 0, Channel 2 (C2) to index 1 etc.), each channel in a CR network has its own index known by nodes. A CR performs the channels-to-slots mapping based on the quality of the channels, e.g., node A has the channel-priority list C2/C4/C3/C1, and therefore C2 is the best channel, C4 is the second best channel etc.

In the first method, *Row-Column* (*RC*) mapping, in the first step (*Step 1* in Figure 3.(a)), a CR (with the channel-priority list C2/C4/C3/C1) selects its map in a row-column manner, where the row number (channel number) is always equal to the column number (channel number). The best A's channel is Channel 2, so it selects a quorum (*row=column=C2=index 1*). The set of elements, represented by (1,1) quorum, maps Channel 2 to $\{1, 4, 5, 9, 11, 13, 14\}$ slots. Each time when a set of elements is chosen, a grid is cut to a sub-grid, together with the already mapped channel, i.e., each sub-grid maintains only the unallocated channels. A set of slots for consecutive channels (according to the quality) is mapped this way till we obtain a 2×2 sub-grid. Note that, each better quality channel has accordingly more assigned slots than a worse quality one. The last two channels are mapped to two slots in a diagonal manner. Analyzing the example (map 2/4/3/1) from Figure 3.(a), Channel 4 map (C4, C4)

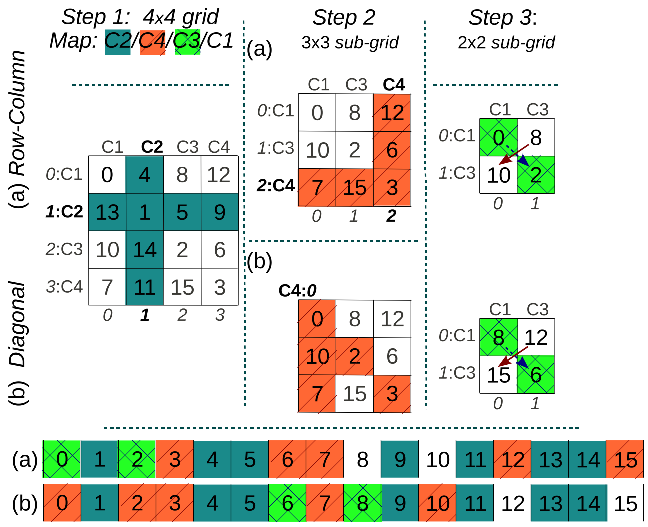

Figure 3. Three steps of the channels-to-slots mapping: (a) Row-Column, (b) Diagonal.

has set of slots: $\{3, 6, 7, 12, 15\}$ (Step 2), Channel 3 is allocated in slot 0 and 2 and Channel 1 to slot 8 and 10 (Step 3).

The second method, *Diagonal* mapping (*CD*), is similar to the Row-Column mapping till we obtain a 3×3 sub-grid. The next channel is mapped (and a sub-grid is cut accordingly) in a column-*diagonal* manner, selecting the first column and the main diagonal, e.g., Channel 4 is mapped to $\{0, 2, 3, 7, 10\}$ slots (*Step 2* in Figure 3.(b)). The last two channels are allocated as in the first method., i.e, Channel 3 is mapped to slots 8 and 6, and Channel 1 is mapped to slots 12 and 15 (*Step 3* in Figure 3.(b)). Note that, a channel map with 3 available channels is an exception following the Row-Column mapping, since the first (the best) channel should always follow a quorum concept.

4.2. Channel mapping optimization

An enhancement of the gQS selection includes attributing the first channels less slots in advantage of the last channels. To do so without loosing the RCP property of the first channel (the reader should note that the property guarantees a RDV if two CRs have the same best channel) we use the *torus* quorum system selection. A torus-based QS [29] is similar to a grid-based QS [15], but normally adopting a rectangular array structure (instead of a $r \times r$ grid) called torus, i.e., wrap-around mesh, where the last row (column) is followed by the first row (column) in a wrap-around manner. The height r (number of rows, i.e. entire column) and width s (number of columns, i.e. entire row) are defined where $n = r \times s$ and $s \geq r \geq 1$. In order to understand the construction of a tQ we present below a standard definition.

Definition 4.1. A torus quorum in a $r \times s$ torus (grid) is composed of $r + \lfloor \frac{s}{2} \rfloor$ elements, formed by selecting any column c_j ($j = 1..s$) of r elements, plus one element out of each of the $\lfloor \frac{s}{2} \rfloor$ succeeding columns using end wrap-around. An entire column c_j portion is called the quorum's *head*, and the rest of the elements ($\lfloor \frac{s}{2} \rfloor$) its *tail*.

In our optimized approach, opt GD^{RC}, (for the sake of the RCP property we optimize the gQS mapping with the row-column selection and diagonal slot distribution) we use a *torus* concept allocating quorum slots for the best channels, but in a $r \times r$ grid, i.e., our torus-in-grid quorum is composed of $r + \lfloor \frac{r}{2} \rfloor$ elements, where a tail is selected in a *forward*-wrap manner (according the standard torus definition)[3]. The remaining elements of the column, which normally are also attributed to the best channels with the gQS mapping, are now equally distributed to the last worst channels. The reader should note that the remaining slots are picked in a backward-wrap manner and assigned to the worst channels so that a worse channel has not more attributed slots than a better channel. Figure 4 presents the opt GD^{RC} method, depicting a tQ forward selection and slot's reassignment for CRs with $C1/C2/C3/C4/C5$ and $C5/C4/C3/C2/C1$ channel priority orders, respectively. The best channel gets seven slots in total, where two remaining elements are assigned to the worst two channels, e.g., Channel 4 and 5 with $C1/C2/C3/C4/C5$ channel ranking list. In the *opt* GD^{RC}*all* algorithm, the full optimization, the second best and following channels must have lesser number of the assigned slots. Therefore, in this example one slot of the second best channel is reassigned to the second worst channel as can be seen in Figure 4. In a partial enhancement, *opt* GD^{RC}*1*, only slots of the best channel are reassigned. Note that while we use *opt* GD^{RC} without *1* or *all* we refer to the full optimization.

Let us analyze scenarios with a larger number of channels. In the scenario with a CR with 10 available channels, while reassigning the best channel slots, four slots are remaining, since a *torus* quorum has $10 + \lfloor \frac{10}{2} \rfloor = 15$ elements, thus four remaining slots, because the number of row plus column elements, in a 10×10 *grid* quorum, equals 19 slots (in the scenario with 20 free channels we get nine remaining slots). With 10 available channels Channel 9 and 10 (assuming $C1/\ldots/C10$ channel ranking list) receive extra two slots from the best channel, and with 20 free channels the last two worst channels get four additional slots and Channel 18 receives one slot.

[3]One should note, that the RCP of a torus-in-grid quorum is guaranteed if and only if nodes select the same column and the same row. In the case of the proposed algorithm this is always the case, since a set of slots of a particular channel is selected from the predefined row and column.

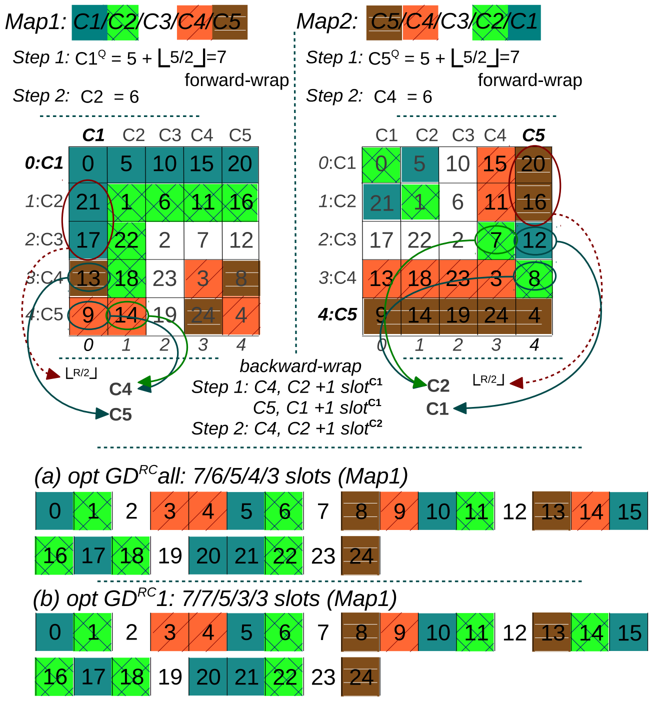

Figure 4. Torus (forward-wrap) way selection in a $r \times r$ grid: (a) opt GD^{RC}all and (b) opt GD^{RC}1.

Summing up, the proposed optimization involves that better quality channels reassign their slots, using a torus instead of grid quorum selection, until there is no worse channel having more slots than a better one. In other words, with an increasing number of available channels the number of extra attributed slots to the last channels increases, bearing in mind a channel ranking.

4.3. RAC^{2}E phase

The RAC^{2}E phase encompasses a cycle (slot) size determination, and MAC process. The protocol relies on an asynchronous operation of the nodes, eliminating the need of synchronization establishment, which is a difficult task in the distributed environments. Moreover, it fosters even an additional randomization among the nodes to ensure a rapid rendezvous on a particular temporary unused channel from the primary system. The operation of the RAC^{2}E phase is illustrated on Figure 5. The reader should note that the figure shows only asynchronism and randomization concept of the protocol cycles, and not exemplary maps of the proposed RAC^{2}E-gQS protocol. Each CR aiming to establish a control channel independently selects a *random* rendezvous cycle duration of Tc_{i_j} (i^{th} cognitive radio, j^{th} cycle). Therefore, although CRs select the same channel map (the same channel ranking lists), they can still have different cycle durations thanks to the used randomization property. This time duration is selected uniformly in the range $[T_{min}, T_{max}]$, where

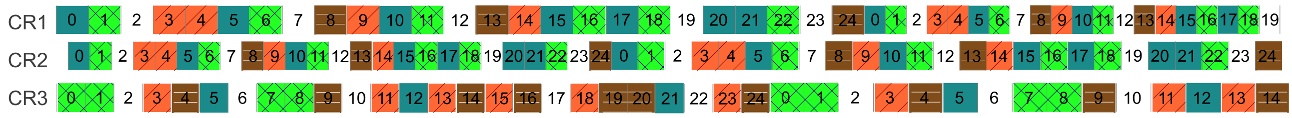

Figure 6. Example of the random cycle duration with three CR nodes, with 5 unoccupied channels, using the optimized RAC^{2}E-gQS protocol: CR1 and CR2 have the same channel order, CR3 has an opposite channel order.

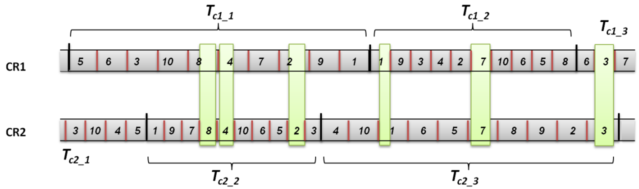

Figure 5. The random cycle duration and asynchronous operation provides overlapping between the both CRs in the free channels (ch_i, $i = 1, ..., 10$).

$T_{min} = T_c - \frac{\Delta T}{2}$, $T_{max} = T_c + \frac{\Delta T}{2}$ and T_c represents the mean rendezvous cycle duration while $\Delta T = kT_c$ is the randomization interval. The chosen Tc_{i_j} interval is further segmented into N time slots, with each slot (having a duration of $\tau_{i\_j} = \frac{Tc_{i\_j}}{r}$) assigned to a particular channel unoccupied by the primary users. As illustrated on Figure 5, the randomization ensures that overlapping at the same channels occurs randomly in wider or narrower time intervals.

Figure 6 depicts an example of the optimized RAC^{2}E-gQS nodes, where CR1 and CR2 have five unoccupied channels with the same channel order, and CR3 has also five free channels, but with an opposite channel order (as shown exemplary in Figure 4). It is easily visible, that there is quite a number of overlaps on different channels, also in the case of the opposite channel orders (e.g., CR2 and CR3).

In each slot interval τ_{i_j}, the CR sends a short beacon message at the beginning and end of the slot to signalize its presence in the channel. These particular times of beaconing are selected since they provide the highest probability of RDV between the CR nodes. In the meantime, between the both beacon messages, the RDV node aims to capture the beacons coming from the other CRs operating on the current channel. As Figure 7 illustrates, the randomization (i.e. asynchronous operation of the both nodes) guarantees that at least one of the beacon messages will be received by other nodes tuned to the same channel at the moment. This justifies the preference of a random Tc_{i_j} duration (Figure 5), which provides a more successful delivery of the beacon messages, in comparison to the synchronous case. A RDV occurs when two nodes are tuned to the same channel and they exchange at least one beacon and one beacon reply message. The condition $\tau' > \tau_{min}$ must be fulfilled for the rendezvous to occur, where τ' is the overlapping duration and τ_{min} is the minimal required time for exchange and processing of both, the beacon and the beacon reply message (Figure 7). Generally, the τ_{min} duration is influenced by the used sample rates of the CR nodes and the length of the beacon and the beacon reply messages. Since there is not much information to transfer, the length of these messages can be in order of few bytes.

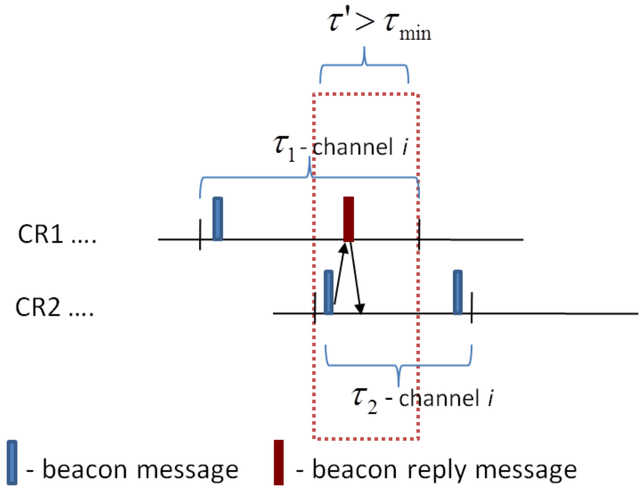

Figure 7. Rendezvous at channel i event.

5. RAC^{2}E-gQS performance evaluation

This section evaluates the RAC^{2}E-gQS protocol with the different channel mapping strategies introduced before. The performance analyses exploit four grid quorum strategies for channel mapping [4, 5] (considering $N = r^2$ number of slots for r available channels): the Row-Column - *RC* and Column Diagonal - *CD* channel mapping, for the both grid forming methods: Pair-on-Pair-Grid (*PoP*) and Grid-Diagonal (*GD*), as well as the UP approach.

The simulation analyses envision a scenario with two CRs aiming for a RDV on a certain common channel. Two main cases are evaluated:

(i) the best case (same), when CRs have the same channel ranking lists, e.g., both have $C1/\dots/C5$ as a priority map for 5 free channels.

(ii) the worst case (opposite), when both CRs have completely different channel ranking list, e.g., CR1 has $C1/\ldots/C5$ while CR2 has $C5/\ldots/C1$ in the case of 5 unoccupied channels.

These two cases are taken as representative examples, since they provide the two extremes of rendezvous performances, i.e., they are the best and the worst case scenarios. Note, that later we also investigate the cases with random channel ranking lists, while comparing the proposed optimization. The detail of the corresponding cases are presented in the respective subsections.

One performance metric of interest in the analysis is the *average number of potential RDVs (channel matchings)* per cycle which is in inverse proportion to the time-to-rendezvous (TTR). Note that the TTR performance for slot synchronized CRs is expressed in slots. In an asynchronous environment, where we evaluate the combined protocol, the TTR is expressed in seconds, because of the nature of the RAC^{2}E protocol. The second evaluated performance metric is the *inter-rendezvous time variance*, representing the variance between two potential consecutive rendezvous, calculated with the Formula 3 and Formula 4:

$$\sigma_{irdv}^2 = \frac{1}{N-1}\sum_{i=2}^{N}((t_i - t_{i-1}) - \mu_{irdv})^2 \tag{3}$$

$$\mu_{irdv} = \frac{1}{N-1}\sum_{i=2}^{N}(t_i - t_{i-1}) \tag{4}$$

where σ_{irdv}^2 is the inter-rendezvous time variance, μ_{irdv} is the mean inter-rendezvous time, N is the total number of rendezvous, while t_i is the time of rendezvous i. For the same number of average potential RDVs per cycle, a higher variance means that channel matchings occur in bursts, leaving longer gaps between bursts, while the lower variance represents the case when channel matchings are more regularly distributed in time. The *lower variance* case is better since it would assure that two CRs going online would not be stuck into the long no-RDV gaps before a successful RDV.

Monte Carlo simulations were made to test the performance of the RAC^{2}E-gQS protocol, for 5, 10 and 20 channels. A total of 10000 trials (RDV cycles) with random start times of the CR nodes were made for each case for statistical correctness. The simulations were performed for a mean rendezvous cycle duration $T_c = 1s$ and duration of $\tau_{min} = 1\mu s$. This τ_{min} duration roughly maps to a case when we have 10Msps sampling rate, 1 byte of beacon and beacon reply message lengths and 4-QAM modulation. Different randomization intervals were evaluated, for k ($k = \frac{T_c}{\Delta T}$) ranging from 1/4 up to 2 with step size of 1/4.

In the first subsection we evaluate the case with slot synchronized CRs using gQS only, followed by the subsection with an analysis of asynchronous CRs using the combined protocol, RAC^{2}E-gQS.

5.1. Slot synchronized CRs using gQS

In order to justify the need of randomization introduced by RAC^{2}E, the grid-QS mapping schemes were tested for a scenario of slot synchronized CRs aiming for RDV. Slot shifts are likely to occur since both CRs do not start the RDV phases simultaneously. Table 3 presents the performances of the grid-quorum schemes in terms of the minimum, the maximum and the average number of potential RDVs per cycle with respect to the slot shifts. As evident slot shifts can cause a high TTR even in the case when both CRs have the same channel ranking lists. The opposite ranking lists and several slot shifts between can result in no RDV between the CRs.

5.2. Asynchronous CRs using RAC^{2}E-gQS

Table 4 presents the average number of potential RDVs per cycle for the RAC^{2}E-gQS protocol, for the same and opposite channel ranking lists of the CRs. It is evident that the case of the same channel ranking lists of the both CRs, results in a higher average number of potential RDVs per cycle than the case with different channel ranking lists. RAC^{2}E improves the RDV performances of the grid-quorum channel mapping schemes, as evident comparing Table 3 and Table 4 results. The channel matching percentage, calculated as the average number of potential rendezvous per cycle divided by the number of slots, is about 52%, 26% and 13.25% for 5, 10 and 20 number of channels, respectively, for the same channel ranking lists case and two times lower for the case with opposite channel ranking lists.

All inspected grid channel mapping methods (PoP^{RC}, GD^{RC}, PoP^{DC}, GD^{DC}), for the particular channel ranking cases and the particular numbers of available channels, provide the same average number of potential RDVs per cycle. The UP mapping method provides the same performances as the grid channel mapping methods when the CRs channel ranking lists are the same. In the case of opposite channel rankings the UP performances differ from the grid-based methods: lower number of free channels results in worse performances compared to the gQS methods; higher number of available channels results in better performances than the gQS schemes.

Although most of the methods experience the same or similar average number of potential RDVs per cycle, they differ in the inter-rendezvous time variance, as demonstrated on Figure 8. It presents the dependence of the variance between consecutive RDVs of both users from the factor of randomization k ($k = \frac{T_c}{\Delta T}$), for the

Table 3. Minimum (Min), Maximum (Max) and average (Mean) number of potential RDVs per cycle for gQS schemes in slot synchronized CRNs; No.c/s stands for Number of channels / slots; List^{Rk} is the channel ranking lists

No.c/s	List^{Rk}	Metr.	PoP^{RC}	GD^{RC}	PoP^{DC}	GD^{DC}
5/25	same	Min	1	3	1	3
5/25	same	Mean	6.52	6.52	6.52	6.52
5/25	same	Max	25	25	25	25
5/25	opposite	Min	0	0	0	0
5/25	opposite	Mean	3.56	3.56	3.56	3.56
5/25	opposite	Max	7	7	7	7
10/100	same	Min	1	3	1	3
10/100	same	Mean	13.28	13.28	13.28	13.28
10/100	same	Max	100	100	100	100
10/100	opposite	Min	0	0	0	0
10/100	opposite	Mean	6.74	6.74	6.74	6.74
10/100	opposite	Max	20	28	30	28
20/400	same	Min	0	3	0	3
20/400	same	Mean	26.645	26.645	26.645	26.645
20/400	same	Max	400	400	400	400
20/400	opposite	Min	0	0	0	0
20/400	opposite	Mean	13.36	13.36	13.36	13.36
20/400	opposite	Max	158	108	160	108

Table 4. Average Number of potential RDVs per cycle for the RAC^{2}E-gQS; No.c/s stands for Number of channels / slots; List^{Rk} is the channel ranking lists

No.c/s	List^{Rk}	PoP^{RC}	GD^{RC}	PoP^{DC}	GD^{DC}	UP
5/25	same	13.042	13.042	13.037	13.043	13.041
5/25	opposite	7.1065	7.1207	7.0994	7.1045	3.6729
10/100	same	26.563	26.557	26.554	26.558	26.559
10/100	opposite	13.409	13.408	13.434	13.356	15.207
20/400	same	53.263	53.243	53.283	53.325	53.296
20/400	opposite	26.543	26.424	26.515	26.504	31.128

cases with the same and opposite channel ranking lists and for 5, 10 and 20 channels. UP achieves a lower variance between potential consecutive RDVs and outperforms the gQS strategies, only for the cases of higher number of free channels and different channel ranking lists, and lower number of free channels and the same ranking lists. The gQS strategies (RC-PoP, RC-GD, DC-PoP and DC-GD) perform better in all other cases. Among the grid quorum strategies, the DC-PoP and DC-DG achieve the lowest variance between RDVs, for the cases with large number of channels, different channel ranking lists and small number of channels, same ranking lists.

Regarding the randomization factor k, it is evident that there is an optimal setting providing the lowest variance between potential RDVs. The optimal k depends on the number of available channels, the difference between the channel ranking lists and the employed channel mapping method (Figure 8).

The fact that gQS strategies encounter somewhat worse performance for the cases of a *larger* number of channels and *opposite* channel ranking lists (e.g., in the case with 20 channels the UP inter-rendezvous variance is ~ 0.008 better than that of gQS) is as expected, since in gQS each better quality channel has accordingly more mapped slots than a worse quality one, but always in a *regular* distributed manner. Although UP channels are also assigned to the same amount of slots as in gQS, the *random* mapping increases an amount of RDVs as well as decreases an inter-rendezvous time variance with a larger number of available channels.
The opposite behavior, i.e., a *large* number of channels but the *same* channel ranking lists can justify this reasoning, since regular mapping is noticeable better approach (e.g., in the case with 20 channels the gQS inter-rendezvous variance can be even ~ 0.05 better than that of UP), but it is favorable to have the same amount of assigned slots for the same channel while having a large number of free channels.
Moreover, while having a *smaller* number of channels in a set (e.g., 5 channels) and opposite channel ranking lists, the regular grid mapping is definitely better (it

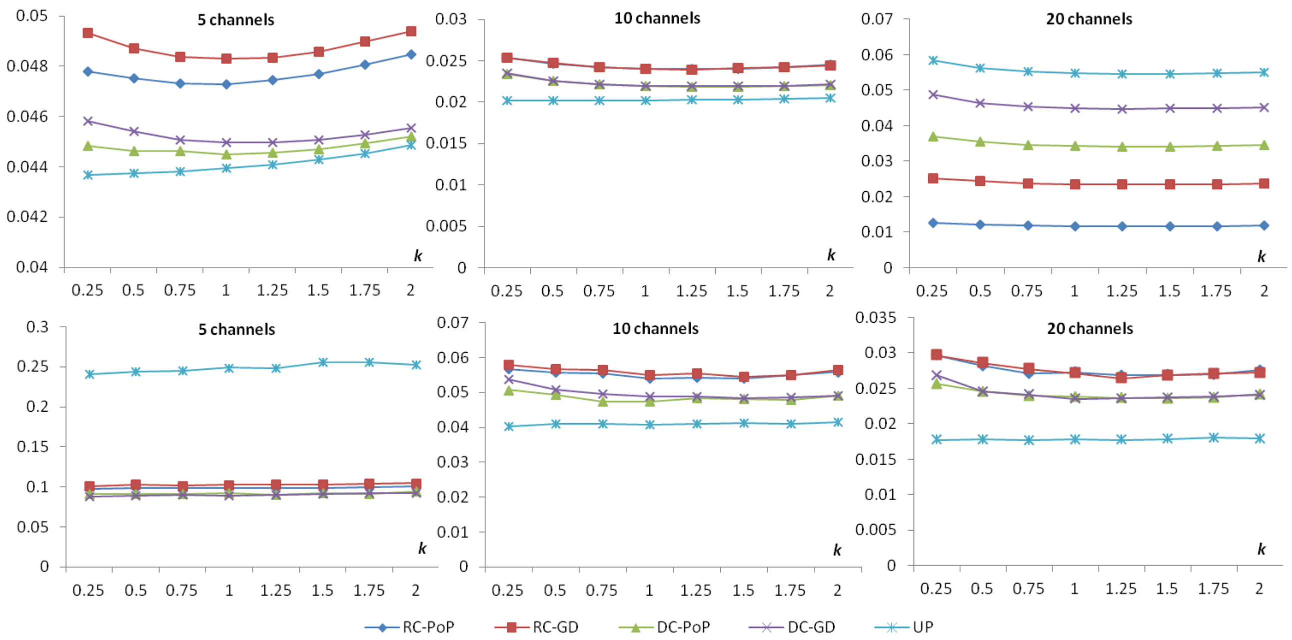

Figure 8. Inter-rendezvous time variance σ^2_{irdv} [sec^2] vs randomization coefficient k, first row: same channel ranking case, second row: different channel ranking case

decreases the inter-rendezvous variance around 0.15), as a difference of the amount of assigned slots of better quality-channels and worse quality-channels is not so drastic as in scenarios with a large number of free channels.

However, since there is indeed a disadvantage of the RAC^{2}E-gQS protocol in the worst case scenario, a large number of channels and opposite channel ranking lists, we apply the proposed optimization (Section 4.2) and compared to the gQS channel mapping. We also show that already a small enhancement (namely, reassigning slots from the first channel only) increases a performance in the scenario in question.

In the next subsection we present the comparison of the gQS channel mapping with its optimization in the slot synchronized case, with the same, opposite and random channel ranking lists, followed by a subsection presenting results of the optimized RAC^{2}E-gQS protocol in an asynchronous environment. The last subsection elaborates on the scenario with asymmetric CRs having heterogeneous channel availability.

5.3. Slot synchronized optimized CRs

Table 5 shows the results of the opt GD^{RC} method with the same and opposite channel ranking lists for 5, 10 and 20 channels in slot synchronized CRNs. As expected in the best case scenario, the RDV performance of the optimized (opt GD^{RC}) approach is degraded in comparison with the GD^{RC} (Table 3), since the number of attributed slots for the best channels is diminished, thus the chance for RDVs on the most frequently visited channels is also diminished. On the other hand, in the worst case scenario the opt GD^{RC} has noticeable improved performance, since worse channels have now more assigned slots, thus more chance to have rendezvous. Since the best case scenario is rather unlikely in CRNs, the opt GD^{RC} can also improve the performance in asynchronous scenarios.

Before going into the details in an asynchronous environment, we show below some results in Table 6 with randomly chosen priorities for 5, 10, and 20 channels in total, ordered as follows:

1. random5: $C2/C4/C3/C1/C5$ and $C5/C1/C3/C4/C2$,
2. random10: $C1 \dots C10$ and $C9/C7/C4/C10/C8/C6/C2/C5/C3/C1$,
3. random20: $C20/\dots/C1$ and $C12/C1/C7/C6/C13/C2/C5/C20/C4/C16/C15/C14/C3/C10/C19/C9/C11/C17/C8/C18$.

As Table 6 shows, the optimized GD^{RC} increases RDV's mean and decreases TTR (measured in slots) of slot synchronized CRs. However, the reader must note, that we can also find random scenarios where the previous version of GD^{RC} has a better performance, since the first best channels are (rather) similar, so the case close to the best case scenario.

Therefore, the further simulation results in an asynchronous environment of the complete RAC^{2}E-gQS protocol are presented in the next subsection.

Table 5. Slot synchronized CRs: Minimum (Min), Maximum (Max) and average (Mean) number of potential RDVs per cycle and average TTR (slots) for the opt gQS scheme; No.c stands for Number of channels; $List^{Rk}$ is the channel ranking lists

Algorithm	No.c $List^{Rk}$	Min RDV	Max RDV	Mean RDV	avg TTR (slots)
opt GD^{RC}	5 same	1	25	5.4	4.66
opt GD^{RC}	5 opposite	0	8	4.6	5.26
opt GD^{RC}	10 same	1	100	11.06	9.05
opt GD^{RC}	10 opposite	0	31	8.96	10.61
opt GD^{RC}	20 same	1	400	21.81	18.34
opt GD^{RC}	20 opposite	0	119	18.25	21.54

Table 6. Slot synchronized CRs, random scenarios: Minimum (Min), Maximum (Max), average (Mean) number of potential RDVs per cycle, and average TTR (slots) for gQS and optimized gQS schemes; $List^{Rk}$ is the channel ranking lists

Algorithm	$List^{Rk}$	Min RDV	Max RDV	Mean RDV	avg TTR (slots)
GD^{RC}	random5	0	8	3.56	6.82
opt GD^{RC}	random5	1	11	4.6	5.48
GD^{RC}	random10	0	28	7.56	13.11
opt GD^{RC}	random10	1	34	9.17	10.92
GD^{RC}	random20	1	164	17.39	23.01
opt GD^{RC}	random20	3	124	19.40	20.62

5.4. Asynchronous optimized CRs

Table 7 depicts the results for the RAC^2E-gQS protocol in the same random scenarios (random5, random10 and random20) as in the synchronized case presented in the previous subsection. In order to see better the difference between GD^{RC} and its optimization, we show also the results while the slots are reassigned from the first best channel only (opt $GD^{RC}1$, example in Figure 4). The reader should note that this small enhancement already improves the average number of potential RDVs per cycle and average TTR.
The results justify the need of having more attributed slots for the worst channels in comparison with the grid QS mapping (GD^{RC}). While allowing of reassignment of all channels (opt GD^{RC}all) the average RDVs significantly increases and the mean TTR decreases.

In the next section we present the results, obtained by using cognitive radios having heterogeneous channel availability in order to verify the protocol behavior in more realistic scenarios.

5.5. Asymmetric channel view

The simulation analyses envision scenarios with two CRs aiming for a RDV on a certain channel while having asymmetric channel views. Four cases are evaluated:

(i) the best case 1 (same1): when CRs have the same channel ranking lists, i.e., CR1 has $C1/C2/C3/C4/C5$ as a priority map for 5 free channels and CR2 has $C1/C2/C3/C4/C5/C6/C7$ as a priority map for 7 unoccupied channels.

(ii) the worst case 1 (opposite1): when both CRs have opposite channel ranking list, e.g. CR1 has $C1/C2/C3/C4/C5$ priority map of 5 free channels, while CR2 has $C7/C6/C5/C4/C3/C2/C1$ in the case of 7 unoccupied channels.

(iii) the best case 2 (same2): when CRs have the same channel ranking lists, i.e., CR1 has $C1/C2/.../C14/C15$ as a priority map for 15 free channels and CR2 has $C1/C2/.../C19/C20$ as a priority map for 20 free channels.

(iv) the worst case 2 (opposite2): when both CRs have opposite channel ranking list, e.g. CR1 has $C1/C2/.../C14/C15$ for 15 free channels, while CR2 has $C20/C19/.../C2/C1$ in the case of 20 unoccupied channels.

All cases are taken as representative examples, since they provide the two extremes of rendezvous performances, i.e. they are the best and the worst case scenarios with a lower and higher number of channels.

The UP method randomly performs the channel mapping with probabilities guarantying that the channels, with respect to their priorities, will statistically be assigned to the same amount of slots per rendezvous cycle as all grid (torus)-quorum based methods.

Table 8 shows the average number of potential RDVs per cycle for the RAC^2E-gQS protocol against the optimized RAC^2E-gQS protocol (optimization for the first channel only, opt $GD^{RC}1$, and all, opt GD^{RC}all), for the *same* channel ranking lists and *opposite* channel ranking lists. It is natural that the case of the same

Table 7. Asynchronous CRs: Average (Mean) number of potential RDVs per cycle and average TTR (seconds) for optimized RAC^{2}E-gQS with optimized slot allocation (a) for the best channel only (opt GD^{RC}1) (b) for all channels (opt GD^{RC}all) in comparison with GD^{RC} from Section 4; No.c/s stands for Number of channels / slots; List^{Rk} is the channel ranking lists

Algorithm	List^{Rk}	No.c/s	Mean RDV	Mean TTR (s)
GD^{RC}	random5	5/25	7.1165	0.1405
opt GD^{RC}1	random5	5/25	8.7188	0.1147
opt GD^{RC}all	random5	5/25	9.1990	0.1087
GD^{RC}	random10	10/100	15.1179	0.0661
opt GD^{RC}1	random10	10/100	17.1173	0.0584
opt GD^{RC}all	random10	10/100	18.3362	0.0545
GD^{RC}	random20	20/400	34.7735	0.0288
opt GD^{RC}1	random20	20/400	35.1575	0.0284
opt GD^{RC}all	random20	20/400	38.8063	0.0258

Table 8. Asynchronous CRs with asymmetric channel views: Average (Mean) number of potential RDVs per cycle and average TTR (seconds) for optimized RAC^{2}E-gQS with optimized slot allocation (a) for the best channel only (opt GD^{RC}1) (b) for all channels (opt GD^{RC}all) in comparison with GD^{RC} from Section 4; No.c/s stands for Number of channels / slots; List^{Rk} is the channel ranking lists

Algorithm	List^{Rk}	No.c/s	Mean RDV	Mean TTR (s)
GD^{RC}	same1	5/25, 7/49	15.9947	0.0626
opt GD^{RC}1	same1	5/25, 7/49	13.8252	0.0724
opt GD^{RC}all	same1	5/25, 7/49	12.5381	0.0799
GD^{RC}	opposite1	5/25, 7/49	5.3882	0.1861
opt GD^{RC}1	opposite1	5/25, 7/49	7.3473	0.1363
opt GD^{RC}all	opposite1	5/25, 7/49	8.6083	0.1163
GD^{RC}	same2	15/225, 20/400	46.8691	0.0214
opt GD^{RC}1	same2	15/225, 20/400	44.3698	0.0226
opt GD^{RC}all	same2	15/225, 20/400	36.9728	0.0271
GD^{RC}	opposite2	15/225, 20/400	15.6793	0.0639
opt GD^{RC}1	opposite2	15/225, 20/400	18.5895	0.0539
opt GD^{RC}all	opposite2	15/225, 20/400	25.3545	0.0395

Table 9. Asynchronous CRs with asymmetric channel views: Average (Mean) number of potential RDVs per cycle and average TTR (seconds) for optimized RAC^{2}E with UP slot allocation optimized (a) for the best channel only (opt UP1) (b) for all channels (opt UPall) in comparison with UP from Section 4; No.c/s stands for Number of channels / slots; List^{Rk} is the channel ranking lists

Algorithm	List^{Rk}	No.c/s	Mean RDV	Mean TTR (s)
UP	same1	5/25, 7/49	15.9926	0.0626
opt UP1	same1	5/25, 7/49	13.7718	0.0727
opt UPall	same1	5/25, 7/49	12.4826	0.0802
UP	opposite1	5/25, 7/49	3.5126	0.2866
opt UP1	opposite1	5/25, 7/49	4.7335	0.2123
opt UPall	opposite1	5/25, 7/49	5.5498	0.1809
UP	same2	15/225, 20/400	46.6890	0.0214
opt UP1	same2	15/225, 20/400	44.1094	0.0227
opt UPall	same2	15/225, 20/400	36.9216	0.0271
UP	opposite2	15/225, 20/400	21.5907	0.0465
opt UP1	opposite2	15/225, 20/400	21.6848	0.0462
opt UPall	opposite2	15/225, 20/400	23.9843	0.0418

channel ranking lists of the both CRs, results in a higher average number of potential RDVs per cycle than the case with opposite channel ranking lists. The opt GD^{RC} method decreases slightly the mean RDV and increases mean TTR while having the same channel ranking lists. On the other hand, it improves significantly the

performance in the worst case scenario.
The next table, Table 9 shows the RAC^{2}E performance but with the UP method. It is clear that the grid (torus) QS slot allocation approach is (slightly) better almost in all cases, except the case with opposite channel ranking list with a higher number of channels, while allocating slots with either GD^{RC} or GD^{RC}1 . However, while applying the full optimization (opt GD^{RC}all) the RAC^{2}E-gQS outperforms all other approaches including the use of the UP method.

These results justify that indeed the slot allocation should be more balanced than in the original version of grid QS mapping (GD^{RC}), and that the optimized RAC^{2}E-gQS improves definitely the performance with different channel ranking lists outperforming RAC^{2}E utilizing the UP method. The best case scenario is unlikely to occur, because in practice due to various propagation effects such as slow/fast fading, scattering etc., the nodes have different view of the surrounding environment. Therefore, the use of the optimized algorithm is to be recommended in situations where a heterogeneous channel view is to be expected. The RAC^{2}E-gQS protocol without the optimization can be used in environments where a homogeneous channel view is to be expected, such as in a controlled lab environment or small scale deployments, where all channels have the same or similar priority.

6. Conclusion

This paper proposes a RDV protocol for CRANs that (i) utilizes the asynchronous and randomness properties of the RAC^{2}E protocol and (ii) the grid quorum channel mapping (gQS-RDV) protocol taking into account channel heterogeneity. We showed that the combination of the RAC^{2}E protocol with the gQS mapping can yield a powerful RAC^{2}E-gQS rendezvous protocol for asynchronous operation in a distributed environment assuring a rapid RDV between the CR nodes. We also propose an optimization of the protocol enhancing its performance noticeable in the case of a large number of channels and different channel ranking lists. The case with asymmetric channel views has also been investigated showing significant improvement of the performance. In our future work, the algorithms will be implemented on a testbed platform with USRP nodes in order to evaluate both approaches in real conditions.

Acknowledgements. We would like to thank Professor Petri Mähönen for fruitful discussions. This work was funded by FP7-ICT NoE ACROPOLIS project [30]. We also thank a partial financial support from Deutsche Forschungsgemeinschaft and RWTH Aachen University through UMIC-research centre.

References

[1] Romaszko, S., Denkovski, D., Pavlovska, V. and Gavrilovska, L. (2012) Asynchronous rendezvous protocol for cognitive radio ad hoc networks. In *Proceedings of the 4th International Conference on Ad Hoc Networks (ADHOCNETS)*. doi:10.1007/978-3-642-36958-2_10.

[2] Akyildiz, I.F., Lee, W.Y. and Chowdhury, K.R. (2009) Crahns: Cognitive radio ad hoc networks. *Elsevier International Journal of Ad Hoc Networks* **7**(5): 810–836. doi:10.1016/j.adhoc.2009.01.001.

[3] Pavlovska, V., Denkovski, D., Atanasovski, V. and Gavrilovska, L. (2010) RAC2E: Novel rendezvous protocol for asynchronous cognitive radios in cooperative environments. In *the 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*: 1848–1853. doi:10.1109/PIMRC.2010.5671630.

[4] Romaszko, S. and Mähönen, P. (2011) Grid-based channel mapping in cognitive radio ad hoc networks. In *22nd Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*: 438 – 444. doi:10.1109/PIMRC.2011.6139999.

[5] Romaszko, S. and Mähönen, P. (2011) Quorum-based channel allocation with asymmetric channel view in cognitive radio networks. In *6th ACM Performance Monitoring, Measurement and Evaluation of Heterogeneous Wireless and Wired Networks Workshop (PM2HW2N)*: 67–74. doi:10.1145/2069087.2069097.

[6] Romaszko, S. and Mähönen, P. (2012) Quorum systems towards an asynchronous communication in cognitive radio networks. *Journal of Electrical and Computer Engineering, Article ID 753541* **2012**: 22. doi:10.1155/2012/753541.

[7] Silvius, M.D., Ge, F., Young, A., MacKenzie, A.B. and Bostian, C.W. (2008) Smart radio: spectrum access for first responders. In *Wireless Sensing and Processing III (SPIE)*, **6980**: 698008–698008–12. doi:doi:10.1117/12.777678.

[8] Theis, N.C., Thomas, R.W. and DaSilva, L.A. (2010) Rendezvous for cognitive radios. *IEEE Transactions on Mobile Computing* **10**: 216–227. doi:10.1109/TMC.2010.60.

[9] Gandhi, R., Wang, C.C. and Hu, Y.C. (2012) Fast rendezvous for multiple clients for cognitive radios using coordinated channel hopping. In *IEEE International Conference on Sesing, Communication, and Networking (SECON)* (Seul, Korea): 434–442. doi:10.1109/SECON.2012.6275809.

[10] Liu, H., Lin, Z., Chu, X. and Leung, Y.W. (2010) Ring-walk based channel-hopping algorithms with guaranteed rendezvous for cognitive radio networks. In *International Workshop on Wireless Sensor, Actuator and Robot Networks (WiSARN2010-FALL), in conjunction with IEEE/ACM CPSCom* (China): 755–760. doi:10.1109/GreenCom-CPSCom.2010.30.

[11] Lin, Z., Liu, H., Chu, X. and Leung, Y.W. (2011) Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks. In *IEEE International Conference on Computer Communications (INFOCOM)* (China): 2444–2452. doi:10.1109/INFCOM.2011.5935066.

[12] Chao, C.M., Tsai, H.C. and Huang, K.J. (2009) A new channel hopping MAC protocol for mobile ad hoc networks. In *Wireless Communications and Signal Processing (WCSP)*. doi:10.1109/WCSP.2009.5371543.

[13] Chao, C.M. and Wang, Y.Z. (2010) A multiple rendezvous multichannel MAC protocol for underwater sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC)* (Australia). doi:10.1109/WCNC.2010.5506099.

[14] Lee, E.K., Oh, S.Y. and Gerla, M. (2010) Randomized channel hopping scheme for anti-jamming communication. In *IFIP Wireless Days (WD) conference*. doi:10.1109/WD.2010.5657713.

[15] Luk, W.S. and Wong, T.T. (1997) Two new quorum based algorithms for distributed mutual exclusion. In *17th International Conference on Distributed Computing Systems (ICDCS)*. doi:10.1109/ICDCS.1997.597862.

[16] Hall, J.M. (1986) *Combinatorial Theory* (John Wiley and Sons).

[17] Bian, K., Park, J.M. and Chen, R. (2009) A quorum-based framework for establishing control channels in dynamic spectrum access networks. In *15th annual international conference on Mobile computing and networking (MobiCom)*. doi:10.1145/1614320.1614324.

[18] Hou, F., Cai, L.X., Shen, X. and Huang, J. (2011) Asynchronous multichannel MAC design with difference-set-based hopping sequences. *IEEE Transactions on Vehicular Technology* **60**(4): 1728 – 1739. doi:10.1109/TVT.2011.2119384.

[19] Bian, K., Park, J.M. and Chen, R. (2011) Control channel establishment in cognitive radio networks using channel hopping. *IEEE Journal on Selected Areas in Communications* **29**: 689–703. doi:10.1109/JSAC.2011.110403.

[20] Bian, K. and Park, J.M. (2012) Maximizing rendezvous diversity in rendezvous protocols for decentralized cognitive radio networks. *IEEE Transactions on mobile computing* **12**(7): 1294–1307. doi:10.1109/TMC.2012.103.

[21] Romaszko, S. (2013) A rendezvous protocol with the heterogeneous spectrum availability analysis for cognitive radio ad hoc networks. *Journal of Electrical and Computer Engineering, Article ID 715816* **2013**: 22. doi:10.1155/2013/715816.

[22] Bian, K. and Park, J.M. (2011) Asynchronous channel hopping for establishing rendezvous in cognitive radio networks. In *IEEE International Conference on Computer Communications (INFOCOM), Mini-Conference* (China). doi:10.1109/INFCOM.2011.5935056.

[23] Shin, J., Yang, D. and Kim, C. (2010) A channel rendezvous scheme for cognitive radio networks. *IEEE Communications Letters* **14**(10): 954–956. doi:10.1109/LCOMM.2010.091010.100904.

[24] Robertson, A., Tran, L., Molnar, J. and Fu, E.H.F. (2012) Experimental comparison of blind rendezvous algorithms for tactical networks. In *IEEE CORAL2012 in conjunction with IEEE WoWMoM2012*. doi:10.1109/WoWMoM.2012.6263760.

[25] Balachandran, K. and Kang, J. (2006) Neighbor discovery with dynamic spectrum access in adhoc networks. In *IEEE 63rd Vehicular Technology Conference, VTC 2006-Spring*, 2: 512–517. doi:10.1109/VETECS.2006.1682877.

[26] Universal software radio peripheral (USRP). URL `http://www.ettus.com`. Accessed March 15, 2013.

[27] Jiang, J.R., Tseng, Y.C., Hsu, C.S. and Lai, T.H. (2003) Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks. In *IEEE International Conference on Parallel Processing (ICPP)*: 257–264. doi:10.1109/ICPP.2003.1240588.

[28] Maekawa, M. (1985) A p n algorithm for mutual exclusion in decentralized systems. *ACM Transactions on Computer Systems (TOCS)* **3**(2): 145–159. doi:10.1145/214438.214445.

[29] Lang, S. and Mao, L. (1998) A torus quorum protocol for distributed mutual exclusion. In *International Conference on Parallel and Distributed Systems (ICPADS)*.

[30] FP7-ICT NoE ACROPOLIS project. URL `www.ict-acropolis.eu`. Accessed November 04, 2013.