

Research Article

Quorum Systems towards an Asynchronous Communication in Cognitive Radio Networks

Sylwia Romaszko and Petri Mähönen

Institute for Networked Systems, RWTH Aachen University, Kackertstraße 9, 52072 Aachen, Germany

Correspondence should be addressed to Sylwia Romaszko, sar@inets.rwth-aachen.de

Received 10 February 2012; Accepted 9 August 2012

Academic Editor: Jaakko Astola

Copyright © 2012 S. Romaszko and P. Mähönen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper reviews quorum systems (QS) from the perspective of cognitive radio networks. Quorum systems were originally developed for and widely used in the scope of operating systems. Recently, quorum systems have been also started to be applied to wireless communications. The objective of this paper is threefold. First, the paper provides survey and guidance on the use of quorum systems. Second, it shows that QS properties provide an interesting alternative towards an asynchronous communication in cognitive radio ad hoc networks (CRANs). Due to properties of quorum systems it is possible to establish CRANs without employing a common control channel (CCC), perfect synchronization, or central controller architecture. QS properties can be efficiently utilized to handle the rendezvous (RDV) problem in CRANs. New RDV protocols must be designed in such a way that there is a guarantee that all nodes meet periodically within reasonable periods of time. Since pseudorandom solutions do not provide this guarantee, systematic approaches are needed such as QSs. Third, we also propose a novel distributed RDV protocol, MtQS-DSrdv, which is based on mirror torus QS and difference set concepts. The proposed protocol guarantees RDVs on all available channels while CR nodes have the same channel set.

1. Introduction

The concept of quorum systems (QSs) is presented [1] in the context of operating systems. Quorum systems are widely used in achieving the distributed mutual exclusion (DMX), for a consistent data replication [2], solving agreement problems [3], or dissemination of information. In the last decade the use of quorum systems has been extended to other applications, especially power-saving (PS) protocols (e.g., to arrange wakeup patterns for PS nodes), node localisation, and gossip protocols [4]. Lately, QS is adopted in order to overcome a rendezvous (RDV) problem or hidden incumbent problem (see, e.g., [5–8]).

A cognitive radio network (CRN) is composed of primary users (PUs) and secondary users (SUs). PUs have the exclusive right to certain spectrum bands, whereas SUs (Cognitive Radio users) have only an opportunistic spectrum access. Therefore, they can temporarily use a licensed band (overlay spectrum sharing [9]), but the appearance of a PU means they must vacate immediately the occupied band.

Hence, link recovery information (and a new determined channel) cannot be circulated over the previously used spectrum because of the PU activity there. A blind date, so-called rendezvous (RDV) problem, in CRANs refers to the ability of two or more nodes/radios to meet each other in the same channel. In Figure 1 Channel 1 is occupied by PU1 (neighbor of CR2) most of the time, and Channel 4 by PU2 (neighbor of CR1). Channel 2 is used by PU3, where both CRs are within its range. (For simplicity we show the situation, where CRs have a common time-slot system. However, we note that this is done without losing generality and, in fact, we have shown also asynchronous operations working in [10]). Within a 16 slots long-time period (period or cycle terms are used interchangeably in this work), CR1 user meets CR2 user on Channel 3 in slot 5 and on Channel 4 in time-slot 12. Users never meet on Channels 2 and 1. The situation depicted in Figure 1 is very optimistic; nodes are able to meet twice. However, while hopping or switching channels randomly it might happen that nodes never meet.

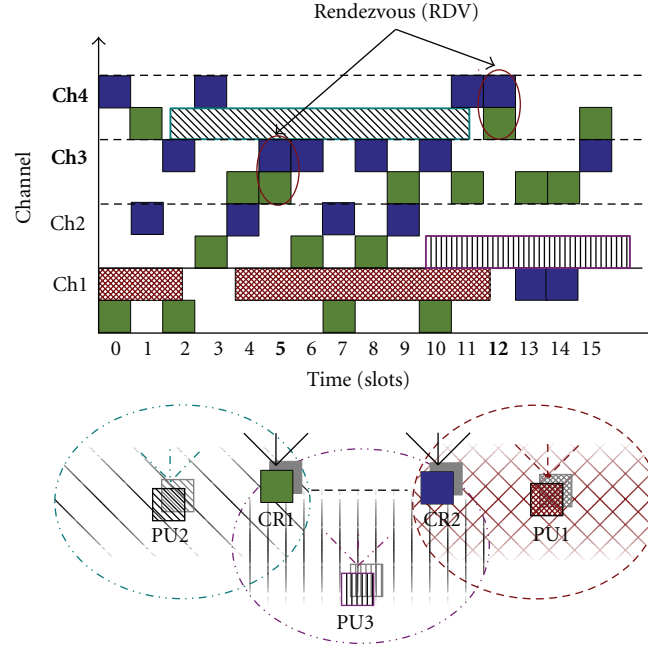


FIGURE 1: Blind date, rendezvous problem in CRANs.

In this paper, we review relevant and interesting quorum-based systems and their properties to show that these techniques are an interesting alternative for establishing asynchronous communication. We also aim to show that a CCC or a strict synchronization between CR nodes can be substituted by other techniques. Within representative techniques, omitting the standard CCC usage is a well-known frequency hopping (FH) technique. An FH technique can be an interesting approach for CRNs, as the probability of interference against PUs decreases. However, a standard FH technique assumes the exchange of frequency hopping sequences and synchronization. This paper investigates different approaches facing these two shortcomings in the CRN context. Finally, a novel rendezvous protocol is proposed, which based on a mirror-torus QS. To the best of our knowledge the torus concept has not been utilized so far in RDV protocols. In this work we also present new approaches to construct a torus quorum in a different way than the standard (forward) one [11], namely, backward [12] and mirror tQS [13], both from our previous work. (Some parts of this research have been published in conference papers [12–15]). The mirror concept allows the construction of torus in a very flexible way.

In Section 2 we present all relevant definitions of QSs and their properties. The rest of this paper is organized as follows. While showing QSs definitions and properties some example applications are already referred in this section. In Section 3 we focus on Power Saving protocols, describing mostly known ones from the application point of view. Afterwards (Section 4), we address a rendezvous issue pointing out the common control channel problems. In the consecutive subsection selected frequency hopping techniques are presented. In Section 5 we focus on QSs

support for dynamic spectrum access where a rendezvous problem is very crucial. Section 6 addresses shortly the grid-based rendezvous protocol. In Section 7 we propose and explain a new RDV protocol, MtQS-DSrdv, and its verification. Section 8 includes a comparison of selected protocols according to the proposed requirements from Section 4.1. The last section concludes this study.

2. Quorum Systems

Quorum-based algorithms have become popular in networking as the main asset of these algorithms is their resilience to node and network failures. Below we present fundamental QS definitions, initially proposed in the context of DMX, followed by more universal ones applicable also to wireless communications. Later on, the most frequently used types of QSs are shortly described.

According to Maekawa's algorithm [1] every process P_i is assigned a request set R_i (called a quorum) of processes, where P_i is in R_i , and for any two processes P_i and P_j , $R_i \cap R_j \neq \emptyset$.

Giving an example (Example I) of Maekawa's QSs [1] for 3 processes let us have the following quorums: $R_0 = \{P_0, P_1\}$, $R_1 = \{P_1, P_2\}$, and $R_3 = \{P_0, P_2\}$. For 7 processes (Z_7) we have $R_0 = \{P_0, P_1, P_2\}$, $R_1 = \{P_1, P_3, P_5\}$, $R_2 = \{P_2, P_4, P_5\}$, $R_3 = \{P_0, P_3, P_4\}$, $R_4 = \{P_1, P_4, P_6\}$, $R_5 = \{P_0, P_5, P_6\}$, and $R_6 = \{P_2, P_3, P_6\}$.

Another definition given by Luk and Wong [16] is as follows.

Definition 1 (Quorum). Let us consider a communication network containing a set of N sites $\{P_1, P_2, \dots, P_n\}$, where P_i is in Q_i , for all $i \in \{1, 2, \dots, N\}$. A *nonempty intersection*

property is defined when $Q_i \cap Q_j \neq \emptyset$ for all $i, j \in \{1, 2, \dots, N\}$. An equal work property is defined $|Q_i| = k$, for all $i \in \{1, 2, \dots, N\}$, such that $k < N$. An equal responsibility property says that P_i is contained in kQ_j 's, for all $i \in \{1, 2, \dots, N\}$.

Finally, a set of sites Q_i is called a *quorum* (Q).

The simplest and usual definition is given in [17, 18].

Definition 2. A quorum system Q under an universal set U , $U = \{0, 1, \dots, n - 1\}$ 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* for all $A, B \in Q : A \cap B \neq \emptyset$.

In the literature one can also find the singleton and majority QS terms:

Singleton Quorum System. a QS under U such that it has exactly one quorum.

Majority Quorum System. a QS under U such that in each quorum it contains more than half of the elements in U .

For instance (Example II):

$Q = \{\{0, 1, 2\}\}$ is a singleton quorum system under Z_3 .

$Q = \{\{0, 1, 2\}, \{1, 2, 3\}, \{2, 3, 0\}, \{3, 0, 1\}\}$ is a majority quorum system under Z_4 .

Aforementioned QS intersection property is already widely used in PS protocols in order to ensure that nodes can wake up during a common time slot (allowing RDV-meeting). However, this property does not guarantee the meeting slot/interval when cycles of the nodes are not aligned. If clock drift occurs, the rendezvous issue poses even more difficulties.

For instance, in the following example (Example III), nodes A, B, and C choose consecutive quorums from the quorum system $Q1 = \{\{0, 1\}, \{0, 2\}, \{0, 3\}\}$ under $U = \{0, 1, 2, 3\}$, that is, A $\{0, 1\}$, B $\{0, 2\}$ and C $\{0, 3\}$. If their clocks are perfectly synchronized, there is no problem as depicted in Figure 2(a).

All three nodes' QSs intersect; thus, nodes can communicate. If the cycles are not aligned (Figure 2(b)), node A meets B in its slot 1, node B can still meet node C in its slot 2, but node A cannot meet node C anymore. Moreover, if there is no cycle and no slot alignment (the latter being an asynchronous case), there is a partial overlap of the slots. A-B have a larger chance to meet in slot 0(A)-2(B), but in 1(A)-0(B) there might be not enough time. Hence, the first step is to find a CCC to communicate. The next step is to find out whether an actual communication is possible within an available amount of time.

2.1. Grid-Based Quorum System. There are different types of quorum systems, within which a *grid*-based system proposed by Maekawa [19] is widely utilized. In this system one organizes sites (elements) logically in a grid. A quorum for

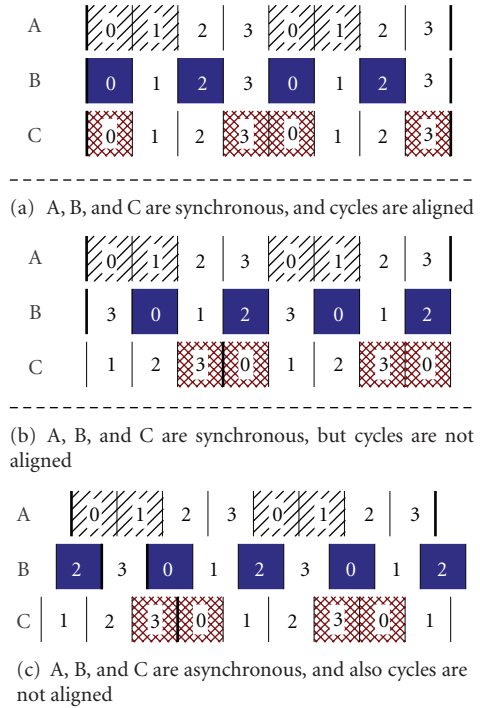


FIGURE 2: Synchronous and asynchronous PS nodes' QS.

a requesting site contains the union of a row and a column that the requesting site corresponds to. Here, the cardinality of a quorum site is twice the theoretical lower bound, that is, $k = 2\sqrt{N}$; however, this algorithm is not well optimized. Luk and Wong [16] optimized the algorithm by organizing the sites in a triangle (in configuration with a row, and with a column), and hence, the quorum size is approximately $\sqrt{2}\sqrt{N}$.

Grid quorum systems are already widely adopted in PS protocols ([4, 18, 20–22]). For example, in [20] PS nodes divide their beacon intervals into groups, where each group includes n consecutive intervals and is organized in $\sqrt{n} \times \sqrt{n}$ array in a row-major manner. Quorum intervals are picked along an arbitrary row and column from this array, where remaining intervals are nonquorum intervals.

Figure 3 depicts an example (Example IV) for 16 slots cycle and three nodes choosing different quorum slots. If the clocks are synchronized (case a) (in cycle and slots) nodes' slots overlap twice, for example, A-B pair in slots 2 and 4, A-C pair in slot 1 and 8, and B-C pair in slots 5 and 10. While cycles are not aligned (Figure 3(b)), all pairs (B's cycle shifted by 2 slots, and C's cycles by 3 slots) meet in slots 0, 4, 8, and 12 (looking from A's cycle view—so starting from slot 0). Figure 3(c) shows that even when clocks are not synchronized (neither cycle nor slot alignment) the A's beacon slots are still covered by the active durations of its neighbors.

In order to maintain the intersection property, the grid cannot be randomly arranged. Chao et al. have defined the grid allocation rules forming a legal grid, where the roles of rows and columns can be exchanged or shifted

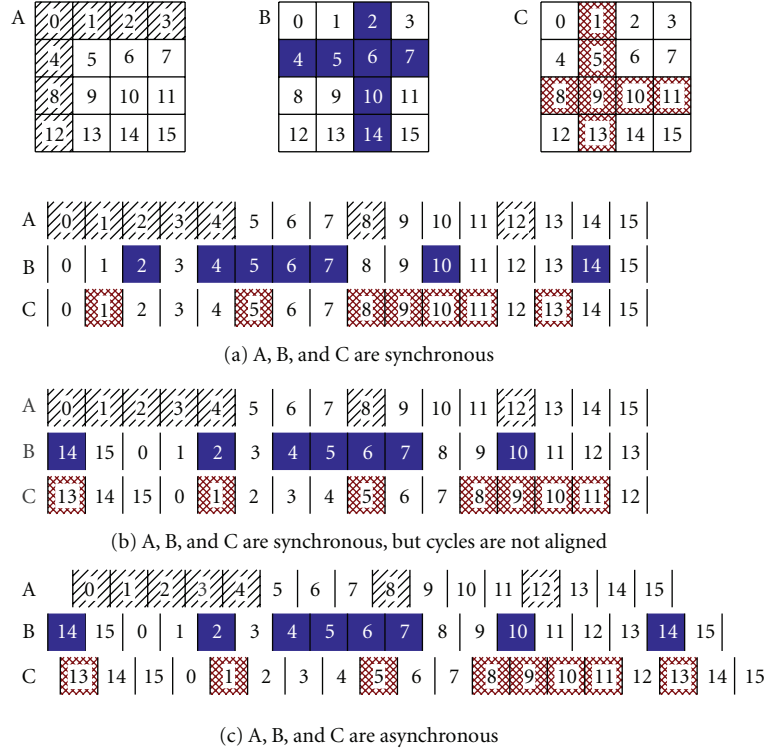


FIGURE 3: Grid QS intersection example.

[23]. In [14, 15] the diagonal distribution of the elements is proposed, called *Grid-Diagonal*, where the numbers are ordered according to the positive diagonal rule; that is, elements are ordered according to

$$f(x, y) = ((y \times n) - ((n - 1) \times x)) \text{ modulo } (n \times n), \quad (1)$$

where $x = \{0, \dots, n - 1\}$ and $y = \{0, \dots, n - 1\}$ as shown in Figure 4. Three nodes select the same quorums as in Example IV, but from the diagonal distribution. In the case when there is synchronization in a cycle and slots (case a), A-B meet in slots 8 and 13, A-C have RDV in slots 4 and 10, and B-C meet in slots 1 and 2. While cycles are not aligned (case b and looking from A's cycle view, so starting from slot 0; B's cycle shifted by 2 slots, and C's cycles by 3 slots) A-B have RDV in slot 4, 7 and 10, A-C meet in slots 4, 7, and 13, and B-C have RDV in slots 1, 4, and 7.

The reader should note that while comparing the standard and diagonal distribution with a cycle shift (case b), there is quite a difference. While using the former arrangement of elements all three nodes meet 4 times but all in the same slots. This might be an advantage, but also a drawback depending on the goal of used quorums. The latter distribution allows to meet 3 times, but one slot is different for all three pairs. Figure 4(c) shows that even when clocks are not synchronized the A's beacon intervals are still covered by the active durations of its neighbors.

In [23], the proposed algorithm makes use of four different grid sizes depending on traffic load. Although the grid size is different, nodes can still intersect (following the defined grid allocation rules). This is a very interesting

property from the communication point of view, where a grid size should be adaptable. However, an n^2 grid array with a larger period size and m^2 grid array with a smaller period may not intersect in a smaller period, although $2m > n$, if a node adopting a larger grid chooses a quorum with the largest row and column index. Figure 5 depicts such problem (Example VI) for 9 (3×3 grid), 16 (4×4 grid), and 25 (5×5 grid) period sizes of four nodes selecting consecutive grids and quorum intervals with and increasing grid size.

Node A having 3 available channels adopts a 3×3 grid selecting the first row and column for its quorum. Node B also chooses the first column and row but from a 4×4 grid. Node C adopts a 5×5 grid having a quorum composed of the last row and column. Finally, node D selects also the last row and column in its 6×6 grid. We can clearly notice that node A cannot intersect with node D in its first period (9 slots). It will intersect only once in the second, never in the third one, twice in the fourth, and again never in fifth periods. Node A does not also intersect with node C in its first and fifth period. Note that not all cases are visible in the figure. Node B does not meet node D in its first period (16 slots!), and only once in its second and third periods. In [15] it has been shown that adopting the grid-diagonal distribution is significantly better (because of a better selection of quorum elements) than the standard distribution of elements; the probability of RDV (the percentage of matches) is larger.

Other interesting grid-based quorum algorithms have been designed in [4]. The first algorithm is a quorum placement algorithm, which maps nodes in a network to elements in QSs. Based on this algorithm, a quorum-based gossip

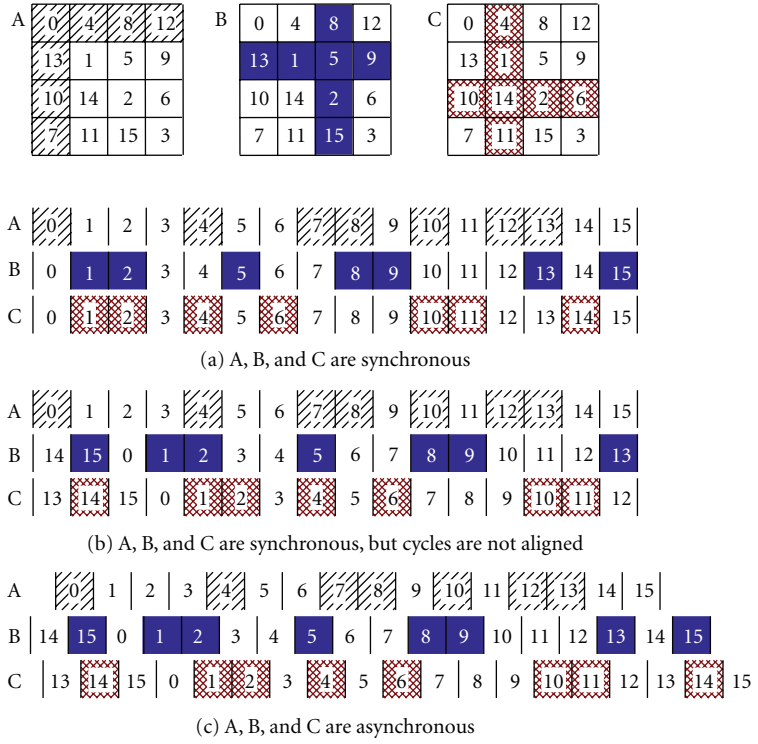


FIGURE 4: Grid-Diagonal QS distribution example.

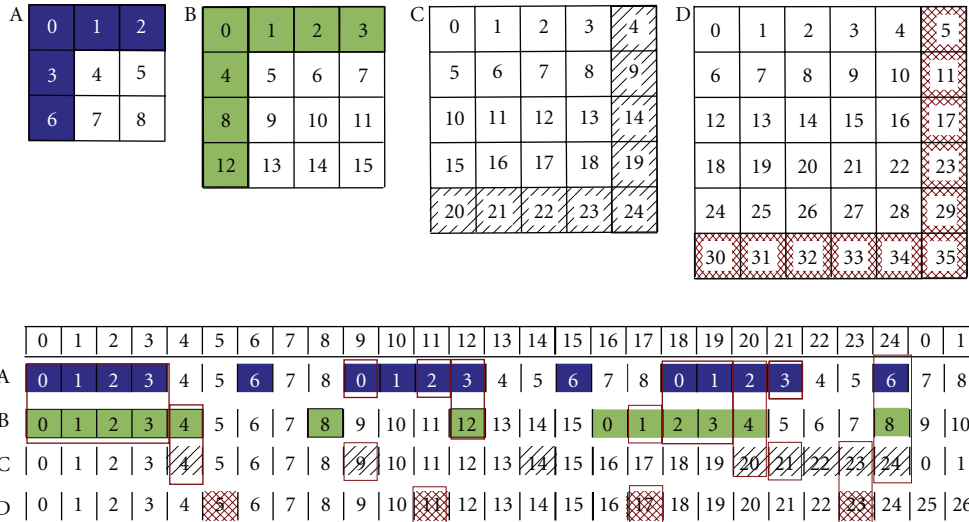


FIGURE 5: Legal quorum intersections with different grid sizes.

protocol was developed which schedules a single task in unreliable networks, that is, with probabilistic termination-time guarantee in the presence of message losses/link failure. The *Grid Byzantine Quorum System* (GBQS) was also presented by Zhang et al. in [4]. It deals with nodes having arbitrary (byzantine) failures due to the destructive behavior disrupting the system, for example, malicious messages. Byzantine QS guarantees that every two quorums have a sufficient number of intersecting elements in order to ensure the majority votes when quorums are accessed by clients.

GBQS could also be used in a medium access scheme, for example, in civil safety, for instance, in rescue systems, where the priority messages have to be delivered (with very high probability) to an intended recipient.

2.2. *Cyclic Quorum System*. Another frequently utilized type of QSs is a *cyclic QS*, presented in [16], which is based on the cyclic block design and cyclic difference sets (DSs) in combinatorial theory [24].

Definition 3 (Cyclic Difference Set). A subset B such that $B = \{a_1, a_2, \dots, a_k\}$ modulo N of Z_n , for $a_i \in Z_n$, is called a cyclic (N, k, λ) difference set under Z_n (k and λ are positive integers such that $2 \leq k < N$ and $|B| = k$), if for every $b \neq 0 \pmod{N}$ there are exactly λ ordered pairs (a_i, a_j) , where $a_i, a_j \in B$ in such a way that $a_i - a_j \equiv b \pmod{N}$.

- (i) If at least one ordered pair (a_i, a_j) exists in (N, k) difference set, then such set is called a *relaxed DS*.

For instance (Example VII), a set $\{0, 1, 2, 4\}$ of Z_8 is *relaxed* $(8, 4)$ difference set as

$$\begin{aligned} 1 &\equiv 1 - 0 \quad 2 \equiv 2 - 0 \quad 3 \equiv 4 - 1, \\ 4 &\equiv 4 - 0 \quad 5 \equiv 1 - 4 \quad 6 \equiv 2 - 4 \quad 7 \equiv 0 - 1 \pmod{8}. \end{aligned}$$

Definition 4. Cyclic Quorum System: given any difference set $B = \{a_1, a_2, \dots, a_k\}$ under Z_n , a cyclic quorum system defined by B is $Q = \{Q_1, Q_2, \dots, Q_n\}$, where $Q_i = \{a_1 + i, a_2 + i, \dots, a_k + i\}$ modulo N , $i \in \{0, 1, \dots, N - 1\}$.

- (i) The cardinality of any quorum in a cyclic quorum system is $|Q_i| \geq \sqrt{n}$.

Definition 5. Cyclic Quorum Sets: a group of sets $\{C_0, C_1, \dots, C_{n-1}\}$ such that

- (1) for all $i \in \{0, 1, \dots, N - 1\}$, i is in C_i ,
- (2) for for all $i, j \in \{0, 1, \dots, N - 1\}$, $C_i \cap C_j \neq \emptyset$, and
- (3) $C_i = \{a_1 + i, a_2 + i, \dots, a_k + i\}$ modulo N

is called a group of cyclic quorums.

- (i) Here, a cyclic quorum set is defined if and only if $B : \{a_1, a_2, \dots, a_n\}$ is a *relaxed* (N, k) difference set.

For instance (Example VIII), there exists a *difference set* under Z_8 , $B = \{0, 1, 2, 4\}$ because it can be generated by the difference of two elements from B for every $b(1, 2, \dots, 7)$. Having B , $Q = \{Q_0 = \{0, 1, 2, 4\}, Q_1 = \{1, 2, 3, 5\}, Q_2 = \{2, 3, 4, 6\}, Q_3 = \{3, 4, 5, 7\}, Q_4 = \{4, 5, 6, 0\}, Q_5 = \{5, 6, 7, 1\}, Q_6 = \{6, 7, 0, 2\}, Q_7 = \{7, 0, 1, 3\}\}$, and it is a *cyclic quorum system* under Z_8 .

In [16] an optimal solution for aforementioned scheme has been investigated and optimal cyclic quorum systems are listed for $N = \{4, 5, \dots, 111\}$.

2.3. Torus-Based Quorum System. Less frequently used, but also a noteworthy system is a torus-based QS [11], which is similar to the grid-based QS [16], but adopting a rectangular array structure (instead of a $n \times n$ grid) called torus, that is, 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 $s \geq r \geq 1$.

Definition 6 (Torus Quorum System). A torus quorum in a $r \times s$ torus (grid) is composed of $r + \lfloor s/2 \rfloor$ elements, formed by selecting any column c_j ($j = 1 \dots s$) of r elements, plus one

element out of each of the $\lfloor s/2 \rfloor$ succeeding columns using end wrap-around. An entire column c_j portion is called the quorum's *head*, and the rest of the elements ($\lfloor s/2 \rfloor$) its *tail*.

Figure 6(a) illustrates an example (Example IX) for three nodes, A, B, and C, choosing three different torus quorums under $U = \{0, 1, \dots, 17\}$, where $r = 3$ and $s = 6$, thus $n = 18$. A is built by picking the third column as its head and 3 randomly chosen slots from succeeding columns. B is formed by selecting the sixth column and its tail from succeeding columns in an end wrap-around manner. C's head is the second column. A and B intersect at slots 11 and 14, B and C at slot 7, and A and C at slots 2 and 9. Later in this work, we call standard tQS as the *forward* tQS, since its tail is selected going forward (to the next columns).

The aforementioned definition is a standard one in the literature [11]. However, the authors in [12] found that it is also possible to construct tQ in a *backward* manner as shown in Figure 6(b). The standard wrap-around torus concept is still followed, but for a tail $\lfloor s/2 \rfloor$ elements one element is selected out of each of the $\lfloor s/2 \rfloor$ preceding columns (c_{j-i} , $i = 1 \dots \lfloor s/2 \rfloor$) using begin wrap-around. Nodes A, B, and C intersect twice in slots 4, 11 (A-C), 9, 14 (A-B), and 7, 12 (B-C).

As presented above we can form a torus walking forward and backward. Following this observation we define the *mirror* torus extension (partially presented in our previous work [13]), which allows to select a tail in much more flexible manner.

Definition 7 (Mirror Torus Extension). A tail of a torus quorum, $\lfloor s/2 \rfloor$ elements, can be selected from any position of column $c_{j+k \cdot i}$ (one element from a column), where $k_i \in \{1, -1\}$ and $i = 1 \dots \lfloor s/2 \rfloor$, in a wrap-around manner. Toruses of the same torus QS need to select elements in the same forward/backward order.

This definition declares, firstly, that if an element was selected from column c_{j+1} , the next element cannot be selected from c_{j-1} , but needs to originate from the next succeeding (forward) column (c_{j+2}) or preceding (backward) column (c_{j-2}). Secondly, the parameter k_i (where $i = 1 \dots \lfloor s/2 \rfloor$) needs to be the same for all quorums of the same torus QS; that is, the direction of the selection needs to be the same. Figure 6(c) shows the selection in a mirror way. Nodes A and B initiate their tail selection in a backward manner, that is, the first element of node A is selected from column c_5 (a mirror of c_1), and the one of node B is selected from column c_4 (a mirror of c_6). Then both nodes select the next tail element in a forward manner, thus from c_{j+2} column (using end wrap-around), which is c_2 in the case of node A, and c_1 in the case of node B. The last element of nodes A and B falls in the same (mirror) column. Nodes meet at slot 4. Note that with more columns, nodes can always alternate the manner, either forward or backward, while picking up elements.

Forward and backward torus QSs use the standard distribution of elements in a torus as it was defined in [11]. However, two other element arrangements can be used next

to the standard one. In our previous work [12] we proposed two different diagonal ways for the distribution of elements. The first diagonal way of forming a torus array is called the *infinite-Diagonal* distribution (inf-Diag), since the way elements are placed (ordered) in a diagonal manner is not limited by the end of row of column by using the modulo operation (%); that is, there is no loop. The distribution of *inf-Diag* torus is shown by

$$f(x, y) = \begin{cases} (y \times (s + 1) - s \times x) \% n & x \leq y, \\ & x = 0 \dots r - 1, \\ & y = 0 \dots s - 1, \\ (s \times (r - x) + (s + 1) \times y) \% n & x > y, \\ & x = 1 \dots r - 1, \\ & y = 0 \dots r - 2, \end{cases} \quad (2)$$

where x is the number of a row, and y stands for the number of a column.

The second diagonal way of forming a torus array is called a *control-Diagonal* distribution (cont-Diag). The elements in a *cont-Diag* torus are arranged according to

$$f(x, y) = \begin{cases} (r \times y - (r - 1) \times x) \% n & x \leq y, \\ & x = 0 \dots r - 1, \\ & y = 0 \dots s - 1, \\ (n - (r - 1) \times x + r \times y) \% n & x > y, \\ & x = 1 \dots r - 1, \\ & y = 0 \dots r - 2. \end{cases} \quad (3)$$

The *standard*, *inf-Diagonal*, and *cont-Diagonal* torus distributions of elements are depicted in Figure 7. While selecting the same head and tail, we obtain the following standard Q elements {0, 1, 9, 11, 18, 21, 27, 31, 36}, inf-diagonal Q elements {0, 9, 10, 11, 12, 13, 18, 27, 36}, and cont-diagonal Q elements {0, 5, 6, 7, 8, 29, 33, 37, 41}. The standard one characterizes with only two neighboring slots and elements rather spread out in the set, but a lot of common elements with the inf-Diagonal set. The diagonal toruses have more neighboring elements, where the second one has elements from the beginning and (rather) end of the set.

Figure 8 depicts an example of five nodes using the backward torus QS and the cont-Diagonal distribution. Each node selects a different head and tail. As it can be seen in the figures, all pair nodes meet at least once.

Figure 9 depicts intersection distributions of a standard (forward) torus quorum, while using standard, inf-Diagonal, and cont-Diagonal ordering of the elements in a 5×9 torus. Note that a tail is selected in the same diagonal way, but all possible combinations are checked, thus starting from different columns and also including a slot offset for every possible offset within a given period. By RDV distribution we mean the amount of RDV in one period (axis x), that is, $1, 2, \dots, s$ RDV in a period, occurs number of times (axis y). Hence, the maximal number of RDV occurrences is 9 (s —the maximum number of columns), since the set has 9 slots.

As one can see from Figure 9, a single RDV occurrence in a period has the smallest number while using *control-Diagonal* torus. The intersection of elements 2, 3, and 4 times in a period is also more often with cont-Diagonal than with the two other toruses, but with RDV 5 times in a period is exact zero, while others have around 160. Although the difference with the intersection occurs 5 times in a period, the *cont-diagonal* torus is still the best, since it has much less RDV only once in a period and more multiple RDV. Moreover, the same situation occurs while selecting a quorum in a *backward* manner (but here note that a particular mapping structure is needed; details in [12]); that is, while mapping channels to parallel quorums, this advantage is increased.

Authors of the paper [18] proposed e-torus system, where the number of intersecting elements in a torus can be adjusted dynamically depending on nodes' mobility keeping in mind that nodes' mobility incurs a higher energy consumption due to an increased number of retransmission attempts. A node with higher mobility can have more intersections (since it is more environment sensitive) than a node with lower mobility.

2.4. Rotation Closure Property (RCP). The *Rotation Closure Property (RCP)* is also a notable mechanism when studying the properties of QS. A quorum system, which satisfies the RCP, ensures that two asynchronous mobile nodes selecting any two quorums have at least one intersection in their quorums.

Definition 8 (Rotation Closure Property). For a quorum R in a quorum system Q under an universal set $U = \{0, \dots, N - 1\}$ and $i \in \{1, \dots, N - 1\}$, one defines the following.

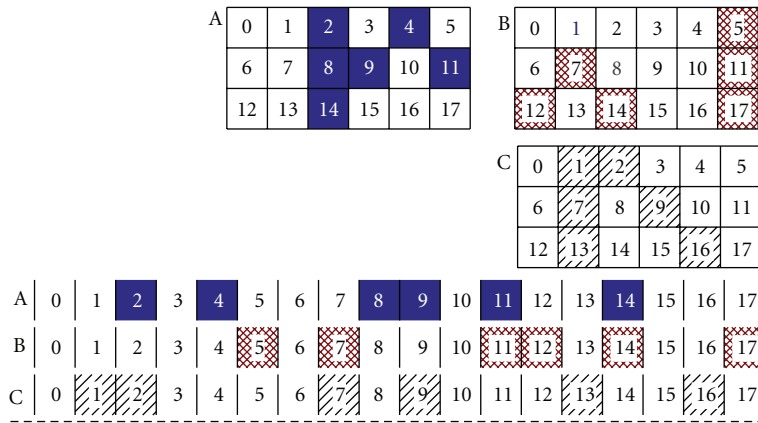
$\text{rotate}(R, i) = (x + i) \bmod N \mid x \in R$. A quorum system Q has the Rotation Closure Property if and only if for all $R', R \in Q$, $R' \cap \text{rotate}(R, i) \neq \emptyset$ for all $i \in \{1, \dots, n - 1\}$.

For instance, in the following example (Example XIII), the quorum system $Q1 = \{\{0, 1\}, \{0, 2\}, \{1, 2\}\}$ under $U = \{0, 1, 2\}$ satisfies RCP; however, the quorum system $Q2 = \{\{0, 1\}, \{0, 2\}, \{0, 3\}, \{1, 2, 3\}\}$ under $U = \{0, 1, 2, 3\}(Z_4)$ has no RCP, since $\{0, 1\} \cap \text{rotate}(\{0, 3\}, 3) = \emptyset$.

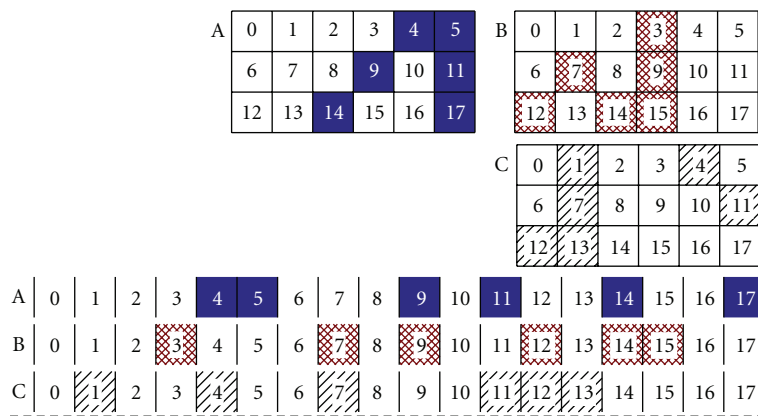
Analyzing the aforementioned Example VII, we can easily notice that, for example, with a time lag of two slots, nodes still meet, $\{0, 1, 2, 4\} \cap \text{rotate}(\{3, 4, 5, 7\}, 2) \neq \emptyset$.

It must be noted that the aforementioned grid (both with the standard [16] and diagonal [14, 15] element distributions), torus (with the standard element distribution [11], inf-Diagonal [12]), mirror torus, and cyclic quorum systems satisfy the Rotation Closure Property.

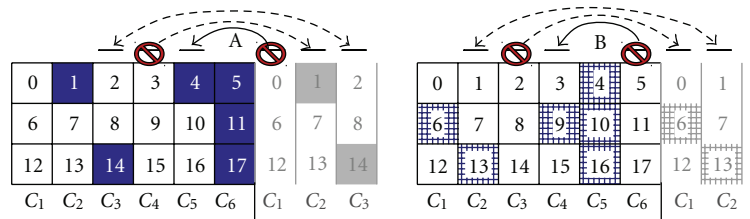
To sum up the aforementioned definitions, one can conclude, that using the intersection property of QS ensures, that nodes which selected quorums from the same quorum system will meet (intersect), if they are *synchronized* in a cycle (frame). However, the intersection is not guaranteed if no cycle alignment is ensured. Hence, the additional solution must be considered (such as cyclic/grid/torus quorum systems or a QS satisfying the RCP).



(a) forward tQS

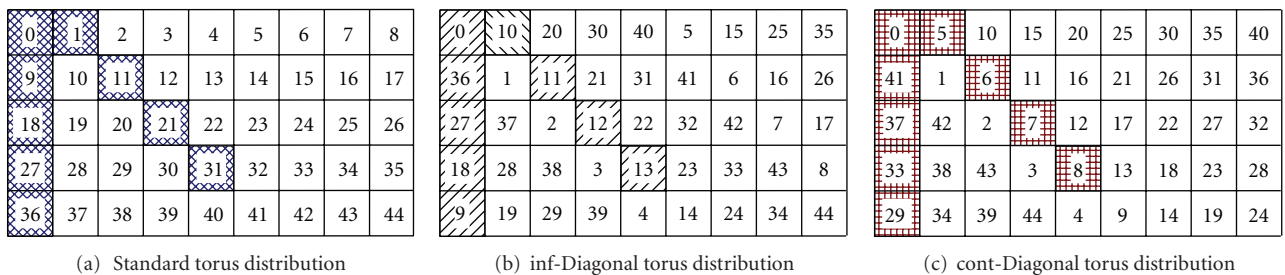


(b) backward tQS



(c) mirror tQS

FIGURE 6: (a) Standard (forward), (b) backward, and (c) mirror tQS.



(a) Standard torus distribution

(b) inf-Diagonal torus distribution

(c) cont-Diagonal torus distribution

FIGURE 7: Standard (a), inf-Diagonal (b), and cont-Diagonal (c) torus distributions.

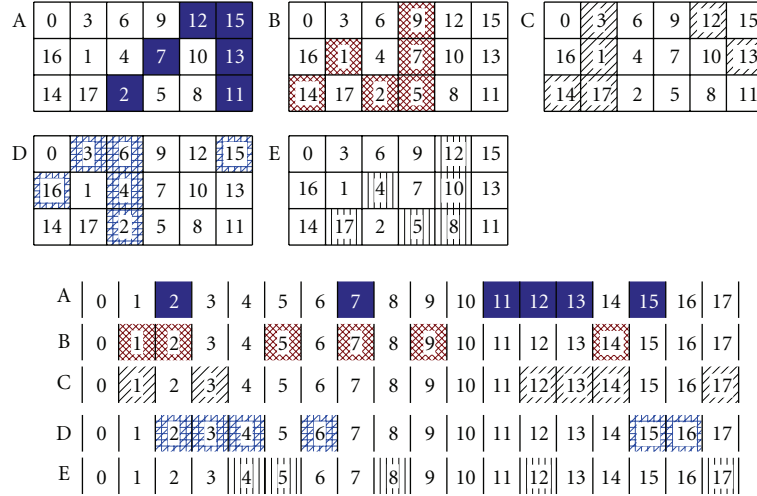


FIGURE 8: Cont-Diagonal element distribution and backward torus QS.

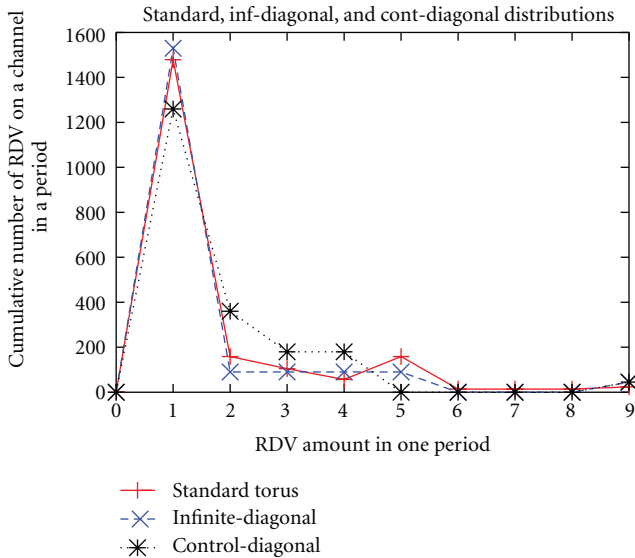


FIGURE 9: Standard, inf-Diagonal, and cont-Diagonal intersection distributions.

It is worth to note that the *Rotation m-Closure Property* (RmCR) exists, guaranteeing that m different quorums intersect even if rotated with different volumes due to a lack of the cycle alignment and/or lack of the slot alignment. This property can be used, for example, to transmit multicast message in asynchronous ad hoc networks.

Definition 9 (Rotation M-Closure Property). For a quorum MR in a quorum system Q under an universal set $U = \{0, \dots, N - 1\}$, one defines the *Rotation M-Closure Property*, for some $m \geq 2$ if for all $Q_1, Q_2, \dots, Q_m \in MR$ and for all $R_1, R_2, \dots, R_m \in U$ and $Q_1(R_1) \cap Q_2(R_2) \cap \dots \cap Q_m(R_m) \neq \emptyset$.

In order to generate QSs satisfying the RmCR, two different approaches were proposed in [25], based on an uniform k-arbiter [26] and Chinese Remainder Theorem (CRT) [27]

quorum system. We do not show here the related definitions, theorems, and proofs since they can be found in [25].

We also list other interesting kinds of QS (properties): majority consensus [28], tree [29], hyperquorum system [30] (cyclic), probabilistic quorums [31], and also heterogeneous RCP, heterogeneous cyclic coterie pair, followed by cyclic quorum system pair proposed in [32, 33]. However, in this study we refer to either frequently used or (easily) applicable in our context.

3. Quorum System Application: Symmetric and Asymmetric Power Saving ((S/A)QPS) Protocols

A considerable amount of work has been done on quorum-based power saving protocols. In the following we provide a nonexhaustive literature review on some of these studies. We start by pointing out the difference between synchronous and asynchronous nodes/protocols since these terms are often loosed in the literature. A *synchronous* protocol is a protocol that requires slot synchronization; that is, the slot boundaries are aligned. In addition, cycle synchronization can be applied; that is, nodes adopt the same cycle. If the slot boundaries of nodes are aligned, but not the cycle, this does not mean that a protocol is asynchronous, only that cycle of nodes is not aligned; thus, there exists a slot lag. An *asynchronous* protocol, on the other hand, requires no synchronization, neither on slot nor on cycle.

Following these terms we list some of the synchronous and asynchronous protocols, where some of them have been classified in the literature as asynchronous, although they are synchronous, but tackling a slot lag problem. Hence, we divide the Quorum-based Power Saving protocols into two categories: (2) *synchronous* QPS [21, 23, 30, 34] relying on the timer synchronization; (3) *asynchronous* QPS [17, 18, 34–37]. The former allows a node to sleep while still guaranteeing the rendezvous of the nodes. The latter prolongs the awake periods to ensure that nodes meet each

other. This approach is useful in the environment (e.g., sensor, cognitive radio, and vehicular networks) where a clock synchronization is often costly or infeasible.

3.1. Synchronous Quorum PS Protocols. In a SQPS protocol, the time axis is divided into *beacon intervals*, and a quorum system defines a *cycle pattern* specifying the awake and sleep schedules during n continuous beacon intervals. Each node using a QPS protocol is required to remain awake only $O(\sqrt{n})$ beacon intervals every cycle, and the overlap of nodes is guaranteed at least once in these beacon intervals. Hence, a node must be awake at least \sqrt{n} beacon intervals to preserve an overlap. The duty cycle can be no less than $O(1/\sqrt{n})$. Considering that an increasing n value increases the delay overhead, the value n cannot be made arbitrarily large. In order to deflect this duty cycle lower bound impact an Asymmetric Cyclic Quorum (ACQ) system is studied in [34]. It is claimed that in the clustered environment there is no need to use a symmetric design. The ACQ system supports the communication between a member of the cluster and their cluster head (CH) and between CHs. As a consequence, each member can have a lighter duty cycle below the traditional $O(1/\sqrt{n})$. However, the system is composed of symmetric quorums (s-quorums), and asymmetric quorums (a-quorums). S-quorums are used to establish symmetric links between gates, CHs, and relays, whereas a-quorums are utilized by the cluster members to communicate with their CH. A-quorums can adopt a small quorum cardinality (quorum size), for example, $O(1)$ -sized. To alleviate troubles during cluster forming period or when a CH is lost, nodes need to follow s-quorums. Since ACQ scheme may also pose heavier duty cycles on nodes using s-quorums, other techniques (such as energy-aware and load-balanced clustering schemes) must be applied. Visible shortcomings (regarding CRNs) of the ACQ scheme are the assumption of the clock synchronization, and neighborhood maintenance. Nodes must exchange the adopted and the current quorum (apart from the cluster information).

Since in existing QPS protocols the quorum cycle *length* is usually restricted to certain numbers (e.g., squares in AQEC [23] or primes in AAPM [38]) the Hyper Quorum System (HQS) is designed in [30]. HQS allows for *arbitrary* cycle lengths (based on the needs of the node, for example, packet delay or power constraints requirements). The concept of the projections of quorums over a plane is used. HQS ensures a shift-invariant intersection (i.e., this quorum scheme is cyclic); hence, it supports any QPS protocol. Two different methods designing HQS with an arbitrary modulo planes (extended grid scheme and difference sets scheme) are also presented.

Another interesting synchronous approach, fuzzy control ([39]) quorum-based energy conserving protocol (FQEC) is presented in [21]. In order to conserve more energy a fuzzy control system is used to adjust the sleep duration dynamically based on experienced latency of incoming and outgoing packets. The quorum concept is utilized to guarantee that two nodes meet each other at some beacon

intervals, while a fuzzy control adjusts the quorum size of each node (sleep duration) dynamically according to its traffic load. Authors in [21] emphasized that while letting nodes to wake up based on the grid-based quorum, the grid size plays an important role in the energy management. In [22] the authors analyze the issues with regards to the thresholds of fixed grid sizes. It is claimed that the optimal grid sizes may be changed over time; therefore, it must be flexible and adaptable. In FQEC two input variables adopted in a fuzzy system are (i) the average transmission delay of the packets received by a node during the last n beacon intervals, and (ii) the queuing delay of the packets waiting in node's buffer. The former is regarded as *history* data (information from finished tasks); the latter is regarded as *future* data (tasks to be processed). A receiver reduces the quorum size to shorten the transmission delay. The quorum size is increased with an increasing idle period (drawback of the protocol visible in bursty traffic simulations). Sender's queuing delay may be reduced to decrease the packet waiting time in the queue. The FQEC protocol prolongs lifetime of the network (comparing with PSM 802.11, AQEC+ [23]) at the cost of an increased transmission latency (circa 56 ms comparing with PSM 802.11; circa 10 ms against AQEC+ [23] in a bursty model but decreased circa 5 ms in a CBR model).

3.2. Asynchronous Quorum PS Protocols. In asynchronous PS protocols there is no assumption on synchronization neither on slots nor on cycle. The Asynchronous, Adaptive, and Asymmetric (AAA) power management protocol is studied in [35] to deal with the timers shifting problem, and asymmetric topology network problem. For the latter problem the clustering is adopted in [34]. In order to solve the former problem a similar solution to that proposed in [34] is suggested, although, in [34] the clock synchronization was assumed. In [35], the same principle (and algorithm) is used, but additional *monitor* beacon intervals (M-BIs) are defined, apart from Q-BIs and S-BIs (not clarified in the paper). Moreover, two types of cycle patterns are defined, the delay-sensitive and best-effort patterns. With the delay-sensitive patterns, nodes also remain awake during the M-BIs, contrary to nodes with the latter patterns. However, an asynchronous approach is somewhat questionable, since in the scope of AAA asynchronous definition, it is assumed that TBTT (Target Beacon Transmission Time) is aligned between stations.

Two asynchronous clock-based PS protocols for Delay Tolerant Networks (DTNs) are proposed in [36]. The protocols are based on a cyclic difference set (CDS) system and have multiple PS levels that can be selected for a different network environment in order to minimize the energy consumption. An exponential adaptive CDS system and multiplicative adaptive CDS are constructed. The former is based on hierarchical arrangements of sets, and the latter is also based on hierarchical arrangements of sets, but a special set of multiplier set is used instead of one exponential set. As a consequence, Rotational Closure Property of the Exponential Hierarchical Design and Rotational Closure Property of the Multiplication Hierarchical Design are defined.

Another asynchronous approach, the consecutive quorum-based PS protocol, is proposed in [37]. The protocol adopts a torus-based [40] quorum system in order to provide Quality of Service (QoS) for applications generating bursty traffic (voice, video applications) and to arrange consecutive intersections in active intervals.

4. RDV Issue in Decentralized CRNs

The frequently assumed common control channel (CCC) in multichannel networks supports the transmission coordination exchange and cooperation between the Cognitive Radio users and facilitates neighbor discovery helping in control signaling, exchange of local measurements, channel sensing, and so forth. However, a CCC existence in CRNs is not without obvious challenges. A CR network is composed of primary (licensed, incumbents) and secondary (unlicensed, cognitive) users. Primary users have an exclusive rights to certain spectrum bands, whereas secondary (cognitive) users have only an opportunistic spectrum access. Hence, they can temporary use a licensed band, but PU's appearance means that they must vacate immediately the occupied band. If we want to use a CCC we must find an uniformly acceptable channel and then take care that this channel is not interrupted over a long period of time or has a regulation that guarantees an existence of CCC. However, the existence of nonregulated CCC is very challenging and sometimes unrealistic in a cognitive environment without any imposed assumptions, since cognitive nodes (can) have a different view of channels occupied by incumbents (while SUs have the same available channel set, we deal with the symmetric channel model (SCHm), otherwise with the asymmetric channel model (ACHm), that is, heterogeneous spectrum availability) and/or other secondary users. Moreover, even if we assume existence of CCC we still need to face a single channel bottleneck problem and a single point of failure.

In [41] the RDV problem is carefully studied in the context of CRNs. The paper provides a taxonomy of the RDV types, relevant RDV spectrum system models, and several different RDV algorithms verified analytically and by means of simulations in terms of Time To Rendezvous (TTR) (TTR is an amount of time, usually measured in slots, within which two or more cognitive radios meet each other once they begun hopping. Since TTR might be not constant, the maximum TTR (MTTR) and expected TTR (ETTR) are used to evaluate the performance. Having a finite MTTRs guarantees RDV) metric. A control channel drawback in CRNs is analyzed shortly, followed by the blind rendezvous problem investigation. However, this study omits quorum systems usage.

In recent work [42] the CCC problem design is also investigated considering the classification, design challenges and schemes, and its applications at network protocol layer. The issues of CCC saturation, robustness to PU activity, limited CCC coverage, and security are identified as major design challenges. The CCC design approaches, for example, sequence-based, group-based, dedicated, and ultra wideband-based, are presented. A relation of the CCC

with radio interface, cooperative sensing, medium access control, and routing are also addressed. In this CCC survey paper there is a reference to a quorum-based CR approach presented in [43], which is an extended analysis of [6]. However, this is only QS-based scheme referred therein.

Another often assumed support for a cognitive spectrum access is a strict coordination or some degree of synchronization between nodes which reduces the rendezvous problem to the minimum. Either a TDMA or Frequency Hopping-(FH-) like access schemes are used with an assumption that nodes can synchronize and coordinate easily, for example, in order to exchange (new) hopping sequences ([44–47]).

The CCC issue has also been investigated in [48], where an extensive survey on MAC strategies in CRNs was conducted. This study emphasizes that the Frequency Hopping Sequence (FHS) strategy is more reliable (since a resource negotiation accuracy does not depend on the status of a single common channel) than using a CCC. However, the authors state that FHS has a drawback because of a tight synchronization requirement among nodes. This is correct, albeit only if a FHS protocol does not exploit another technique supporting rendezvous. Similarly in [49] it is also argued that sequence-based design has little adaptation to PU activity (see also [50, 51]). There are many different sequence-based approaches that are more or less adaptable to the PU activity, where the adaptability to the PU activity depends on the design and objective of an algorithm. In [49] it is also noted that a sequence-based design incurs a high control overhead due to broadcast messaging. However, it is assumed that SUs must cooperate to enhance the detection accuracy of PU activity and coordinate a free spectrum access by control message exchange. Thus, the CCC existence is imposed in [49]. However, the authors of [25] show that there exist techniques allowing for multicast messaging in asynchronous ad hoc networks. It has been also argued that the broadcast issue is not a major problem in CRNs, since, anyway, cognitive radio characteristics can never assure an optimal broadcast solution which is also not an objective of CRNs. Moreover, in CRNs there is no optimal channel allocation approach. CCC, sequence-based or central controller approaches will always have some merits and shortcomings, since they have to deal with an uncertain and dynamic cognitive radio environment. Thus, in our opinion, it is unlikely that any of the methods are superior in all possible application scenarios. Hence, one has to study performance and relative merits based on scenario analysis before selecting a right approach.

4.1. Channel Hopping Requirements in CRNs. Frequency hopping techniques give an interesting alternative to strive with problems encountered in a multichannel environment. Further in this study we will mostly focus on FH techniques with a reference to the asynchronous RDV issue (see also [48]). However, before going into details of numerous FH approaches, a few essential requirements must be noted that are often omitted or neglected in some of the schemes.

- (i) *1st channel hopping cognitive (CHC) requirement:* an overlap between the periods (here period or

cycle terms are used interchangeably) of any pair of nodes (overlap between their channel hopping sequences) should guarantee a very high probability of rendezvous.

- (ii) *2nd CHC requirement*: nodes can rendezvous in more than one channel within a sequence period. This is essential in CRNs, due to a possible sudden unavailability of a channel due to an unpredictable appearance of PU signals. It must be noted that it is desirable to guarantee RDVs on different channels (every possible channel), for example, instead of two distinct channels.
- (iii) *3rd CHC requirement*: a TTR value needs to be small and bounded.
- (iv) *4th CHC requirement*: because of the unpredictable nature of CRNs, a proposed scheme should anticipate the clock drift problem; thus, it should not rely on any tight synchronization or coordination.
- (v) *5th CHC requirement*: any global or neighboring information/dependencies should be minimized or avoided.
- (vi) *6th CHC requirement*: symmetric and asymmetric channel views (heterogeneous spectrum availability) must be handled since a sudden incumbent presence is critical in CRNs. Only a few papers really elaborate on this issue.

4.2. Frequency Hopping in Commercial Systems. Frequency hopping is used in the context of Global System for Mobile communication (GSM) cellular systems, Bluetooth, and many military radios among others. In GSM systems it can be used optionally in order to mitigate fast fading and cochannel interference. In the GSM systems frequency is changed each 4.17 ms according to a prespecified sequence. In the Bluetooth systems, hopping is performed every 0.5 ms and a sequence is chosen among several prespecified hopping sequences. Moreover, Adaptive Frequency Hopping (AFH) has lately been designed for the second generation Bluetooth in order to adapt to the environment (by recognizing fixed interference sources to exclude them from the frequency hopping list). Dynamic Frequency Hopping (DFH) [44] has also been proposed in IEEE 802.22 draft standard [52] (a cell is able to hop over a set of channels). However, all these solutions [44, 53] require synchronization, strict coordination, or a good communication between cells (neighbors) in order to avoid an overlap of a newly chosen channel by different cells.

4.3. Frequency Hopping for Cognitive Networks. A number of FH schemes assume a degree of synchronization and thus not necessarily directly comply with the 4th CHC requirement. For instance, in [45] synchronized SUs determine their pseudo-random (PR) hopping sequence (fixed for a given secondary user) based on their unique ID and also share the same hopping sequence generating algorithm. Each node needs to broadcast its own hopping sequence over an unused channel in order to inform neighbors about its sequence. In [46] a dynamic parallel rendezvous multichannel MAC

protocol for synchronized CRNs is proposed, in which SUs modify their own distinct hopping sequences according to the datarates of the available channels, in a datarate-aware manner.

However, in cognitive networking, instead of the use of CCC, strict synchronization, or coordination between nodes, there is a need to have a robust approach that can handle link breakage due to unpredictable PU activities.

The first step towards this goal is visiting channels in a random fashion, called also a *blind random* RDV [54] or PR hopping sequence [55]. However, in such method the time to rendezvous (TTR) is unbounded (3rd CHC requirement) and there is no guarantee that nodes will actually meet (1st CHC requirement). Therefore, enhanced approaches have considered the use of *patterns* or *sequences* determining the order with which nodes visit potentially available channels. While having a nonorthogonal hopping sequence with specific patterns of *periodicity*, it is possible to establish an upper bound to the TTR, determine the order of priority channel, and to increase the expected TTR (compared to a blind method). The well-known work in this realm is [50]. In this work DaSilva and Guerreiro proposed to use sequences constructed in such a way that they minimize TTR also considering the possibility when cycles are not aligned to each other (a time lag). A permutation of n channels is utilized when selecting a hopping sequence. The ability of channel prioritization is also considered while there is a need to use more frequently those channels that are less occupied by PUs. Moreover, a primary user appearance is taken into account in the proposed algorithm. When an incumbent is detected, a sequence is updated, but there is no guarantee that a rendezvous will occur (1st CHC requirement).

In the most hopping sequence techniques nodes must hop together in synchronized manner. This is not always feasible in a distributed CR environment. In order to solve this problem the authors of [55] propose an adaptive multiple-rendezvous control channel which, in contrary to [50], uses adaptively chosen sequences when PU's activity is detected. In [55] nodes hop in an asynchronous manner, unknown to their neighbors before their RDV. Hence, the common hopping sequence overhead is reduced. Based on the sensing information, a *ranking table* of available channels is constructed based on a primary user activity. Nodes generate a pseudo-random hopping sequence, which is mapped to the ranking table in order to obtain an adaptive hopping sequence. This mapping increases the probability to have an RDV on a channel with a low PU activity and to decrease the time to attain an RDV. However, the synchronization of nodes is imposed after meeting on a common channel. There are a number of assumptions which are not easy to ignore in CRNs, for example, collisions of control packets of SUs with PU packets.

In [56], the multi-channel, asynchronous rendezvous problem is addressed analytically and experimentally with four cognitive radio nodes. The conclusion has been drawn that RDV does not occur for multi-channel cases without slot repetitions for any value of a transmit probability.

An adaptable scheme to PU's activity was proposed in [57]. Although authors present also a synchronized approach, the scheme merits attention thanks to the more elaborate way of creating a dynamic channel hopping sequence. Instead of using a simple pseudo-random manner, it utilizes of a cyclic sequence composed of small subcycles (l-cycle) forming a big cycle (b-cycle). Two different seeds (a hopping seed and a shifting seed) determine the hopping sequence of subcycles. The number of channels, n , must be a prime number and the boundary of the subcycle should be synchronized for operation. Complying with this assumption guarantees that nodes meet at least once during a subcycle. A desynchronization operation is envisaged in the case when a contention is formed by selecting the same channel due to the same CH parameter set. According to the algorithm, SUs not communicating with each other meet less often. Hence, multiple rendezvous is possible; that is, neighboring SUs transmit in parallel without interfering with each other when hopping to different channels.

Other more sophisticated approaches, such as Ring-walk based channel-hopping algorithms, are proposed in [58] and analyzed in symmetric and asymmetric channel models (6th CHC requirement). In these algorithms each channel represents a vertex in a ring in such a way that the CH sequence generation is equivalent to visiting vertices in the ring. Each node walks on the ring in a clockwise direction visiting vertices with a preassigned velocity. In each time slot, a node stops at a vertex (channel), and in the next time slot it remains or moves to the next vertex depending on its assigned velocity (each user has a unique ID which could be used to design its distinct velocity). Since nodes have different IDs (i.e., different velocities) RDV is guaranteed, because nodes with a larger velocity catch up nodes with a lower velocity. However, it is assumed that nodes know the largest possible ID, that is, the network size n which also influences the period size composed of n time slots. (thus, 5th CHC requirement is not fulfilled). Moreover, depending on the ID, nodes stay longer or shorter (the user with larger ID walks faster) on each channel impacting the created sequence. However, an ID is a bad factor for such decision, since other factors are really influential (e.g., PU's channel occupancy or traffic load). Moreover, the fairness is neglected in [58], since nodes switch channels faster or lower depending on their ID.

We conclude this section by describing Jump-Stay-(JS-) based channel-hopping algorithm proposed recently in [59] for the blind RDV problem. It is claimed that the algorithm guarantees rendezvous without exploiting the time synchronization, both in the symmetric and asymmetric channel models (SCHm and ACHm, resp.). The JS algorithm generates channel hopping patterns (a jump pattern followed by a stay pattern) in rounds based on three tuples, (2) the smallest prime number P greater than the number of available channels M , (3) a step length r_0 factor modified (each round) using a modular arithmetic with respect to, either P in SCHm, or M in ACHm, and (3) an index i_0 being a constant in SCHm and modified using a modular arithmetic with respect to P in ACHm. In SCHm with 2 nodes, the maximum TTR (MTTR) is equal to $3P$, and in ASCHm

$MTTR = 6MP(P - G)$ with G representing the number of common channels. The advantage of the algorithm and the algorithm's TTR is compared with the TTR of algorithms investigated in [41]. If we go through an example with only 2 channels ($M = 2$), then we have $MTTR = 9$ in SCHm and $MTTR = 72$ in ASCHm (assuming one common channel). With three channels, the MTTR increases significantly to 15 in SCHm, and 270 in ACHm (assuming two common channels). TTR might take a considerable amount of time, especially in ACHm, but also in SCHm when the number of channels is larger. Therefore, it could happen that nodes do not rendezvous finally, since meanwhile the channel availability can change. A possible sudden change of the channel set is not investigated in this work, which might be a potential drawback of this approach. Moreover, the JS algorithm is not compared with quorum-based techniques that ensure a smaller TTR.

5. Quorum-Based Multichannel MAC Protocols

We will next focus on quorum-based systematic approaches. We start by reviewing quorum-based MAC protocols, and then discuss their applicability to a design of CRNs.

The Cyclic-Quorum-based multichannel MAC protocol is proposed in [60] in order to support *multirendezvous*, that is, where multiple transmission pairs can accomplish simultaneous handshaking. The protocol makes use of the channel hopping as a foundation, selecting hopping sequences individually (without signaling via any control channel). It also utilizes the cyclic QSs to accomplish channel allocation, to guarantee nodes rendezvous, and to give equal opportunity for nodes to transmit and receive packets. In each cycle, time slots are divided into *default* slots (Rx: waiting for transmission requests) and *switching* slots (Tx: (channel) switching to intended receivers). A cyclic quorum and difference set are utilized to identify node's default slots and channels, also based on node's ID. Two nodes adopting the same cyclic quorum have no overlap of default or switching slots. It might happen that nodes having different default channels may not meet. In such case a temporary substitution of the default channels with the switching channels could be envisaged. This is a potential shortcoming of the scheme. Moreover, the protocol has other drawbacks when considering cognitive radio networks. For example, clock synchronization of nodes is assumed and each node must also know the ID of its one-hop neighbors (which demands an initial communication).

In order to reduce the collision probability in Underwater Sensor Networks (UWSNs) the concept of cyclic QSs has also been adopted in a distributed multiple-rendezvous multichannel MAC protocol presented by Chao and Wang in [61], also addressing the missing receiver problem. This protocol has a similar algorithm and shortcomings as the one proposed in [60], that is, need to have clock synchronization, knowledge of neighbourhood IDs is required, and rendezvous is not guaranteed if nodes adopt different default channels.

A quorum-based RDV channel hopping (QRCH) medium allocation scheme in a hostile and jamming environment is presented in [5]. A quorum concept is utilized for a RDV guarantee, and to avoid using an explicit prekey establishment to achieve a fast and resilient communication. In QRCH elements from the selected quorums are mapped into channel indexes, and nodes generate a sending and receiving hopping sequence. The quorum cardinality k determines the length of the time period, k^2 time slots—where one period consists of $|k|$ frames each of which consists of $|k|$ slots. The protocol is evaluated in the presence of different jamming attacks. However, authors do not explicitly state whether the protocol can handle asynchronous clocks (3rd CHC requirement) and the 6th CHC requirement. The 2nd CHC requirement is also not guaranteed by this method. However, it is interesting to note that the asymmetric channel view (but with only the same number of channels) is implicitly addressed.

5.1. Quorum-Based Channel Hopping Study in CRNs. There is only a limited amount of work that specifically exploits quorum-based methods in combination with FH techniques in a CR environment. The proper use of quorum-based schemes can be advantageous compared with other methods since they can satisfy the 1st, 3rd, and 4th CHC requirements. Moreover, they can also facilitate the compliance with the 2nd CHC requirement.

In [8] the hidden incumbent problem in the context of IEEE 802.22 networks is addressed. (Hidden incumbent problem refers to the situation in which a consumer premise equipment (CPE) being in the incumbent transmission range (TR) may not be able to decode the base station (BS) signal due to the strong interference from the incumbent signal. However, CPEs cannot report the existence of the incumbent since their transmission will interfere to the incumbent. Thus, BS fails to detect the incumbent presence). To tackle this problem the standard defines an explicit outband signaling (EOS) used by BS for periodic broadcast of outband control signals. Normally, the set of candidate channels for CPE is the same as BS's set of candidate channels. However, it is possible to reduce the size of such set, and thus lowering the control overhead, by making use of QSs to construct it in such a way that at least one common channel can be used for uplink/downlink communication [8]. The authors conclude that the use of a majority QS can reduce the size of the candidate channel set to $(n+1)/2$ (thus by half); a cyclic QS can decrease the size of the candidate channel set to \sqrt{n} (thus by square root).

Three different approaches for quorum-based channel hopping (QCH) in CRNs are proposed in [6]. The first two synchronous approaches under the assumption of global clock synchronization focus either on TTR minimization (M-QCH approach) or an even distribution of the rendezvous points in terms of time and channel (L-QCH approach). The proposed synchronous M-QCH system, which is based on a majority cyclic QS, can support m rendezvous channels (where $m \in [1, N]$) having the lowest MTTR among QCH systems. L-QCH addresses a control

channel bottleneck problem, where ideally, a CH protocol should spread out the RDV over all channels evenly. This objective might be somewhat questionable in CRNs, since some channels can be frequently used by SUs, while others can only be occupied by SUs from time to time due to a high probability of the PU activity. The third solution is an asynchronous method. The proposed quorum-based channel hopping system is constructed in such a way that it also satisfies the 2nd CHC requirement. In [6] the authors state that the rendezvous spread out over a larger number of distinct channels will diminish the link breakage probability caused by an inability to exchange control packets. However, the algorithm authors study is designed for the systems with *only* two intended distinct RDV channels (randomly chosen form a set of the channels). In [6] a comparison of a number of studies [50, 54, 62] can also be found in terms of a degree of overlapping, MTTR, load, and asynchronous operation.

In a recent paper [43], a somewhat broader analysis for these three quorum-based approaches is presented along with studying an additional asynchronous approach, A-MOCH, which enables the rendezvous on every channel. A-MOCH is not based on QSs, but it is designed in such a way that the RCP of QSs is satisfied; therefore, we find it relevant for our review. In this method each node generates two different sequences: the alternative channel hopping (CH) sequence for a transmitter and the default CH sequence for a receiver. The former is based on a Latin Square (LS), where numbers in a table ($n \times n$, where n —maximum number of channels) are arranged in such a way that each number occurs exactly once in each row and each column. The default CH sequence is based on an Identical-Row Square (IRS), where each row in a table ($n \times n$, where n —number of channels) is a permutation of different integers (n channels). It is shown that with these two CH sequences in A-MOCH, the degree of overlapping is n (while having n channels) in the period of n^2 , where MTTR equals $n^2 - n + 1$. On the one hand, this is an interesting approach since there is the rendezvous guarantee on n channels. On the other hand, this approach has shortcomings imposed by the character an IRS array. Receivers have to jump constantly to a different channel since their sequences have no neighboring slots. It might also happen (cf. [43]) that a transmitter will select such LS, which also forces constant channel switching behaviour.

In [63] a similar approach to [43] has been proposed where a transmitter (instead of a receiver) adopts IRS arrays. A receiver creates its sequence based on a concept of LS, since one number can occur exactly once in a column, but not necessary exactly once in a row. A receiver in A-MOCH [43] and a transmitter in ACH [63] switch constantly the channel, which as a consequence diminish their chance to meet when nodes are asynchronous since an overlap probability between unaligned slots might be too small for a successful RDV.

In both approaches ([43, 63]) the *channel switching time* cost must be accounted, either for a transmitter or for a receiver (or both if the LS sequence does not contain neighboring slots for the same channel), due to the maximum frequency of channel switching. Therefore, a flexible spectrum handoff must also be considered. (A proactive or reactive spectrum handoff should be adapted to the

type of application and network environment [64]. Delay-sensitive applications should have a reduced switching time (proactive switching is desirable through the prediction of the PU activity); however, in energy constrained devices, for example, sensors, reactive switching must be considered). A related and very important issue is the energy waste imposed by the constant hopping, since nodes will visit all channels.

In [7] a multichannel MAC protocol is proposed dealing with a lack of synchronization and CCC existence thanks to the generation of the same hopping sequences by all users based on multiple number of different sets (DSs). The incident vector of a DS, complementary set, and RCP are defined in this study in such a way that RCP ensures that a (n, k, λ) difference set of Z_n (k -cardinality of DS) and any of its shift sets (as in cyclic QS) have λ overlapping elements in a cycle of n , where λ/n denotes the rendezvous probability. It can be shown that regardless of the synchronization two nodes having the same hopping sequences generated from DSs can always rendezvous with a certain probability. Using DSs and complementary ones (satisfying an empty intersection criterion) the channel set is mapped to these DSs. However, a different channel view is not considered in this study, which can impact the performance of the algorithm in CRNs. Moreover, all nodes adopting the same hopping sequence can make them more exposed to collisions. Some nodes can join the network at different time instants (utilizing the RCP). However, the network joining can also happen at the same time instant; that is SUs can rendezvous at a particular channel with a certain probability dependent on the number of contenders. The 2nd CHC requirement (guarantees that nodes meet at least twice on different channels in one period) is discussed by *Criterion 1* while designing a common CH sequence. Overall, DSs must be very carefully selected in order to follow this criterion. The readjustment of the channel hopping sequence might be needed since some channels might be inactive during a long period of time.

In [65] sequenced-based protocol has been implemented on USRP boards and evaluated in terms of the time of the first encounter between two SUs and in terms of the time for encounter on all channels. The protocol itself is partly based on quorum-based algorithms [18, 66]. N sequences are formed (N stands for the number of channels) in such a way that the length L of each of them is a prime number of the power of a prime number so that L_i and L_j are coprime. Nodes can discover each other on all N channels within a bounded time, if they select different of aforementioned hopping sequences. Using the same hopping sequences implies a RDV possibility only on one channel (later in the verification section we will show an example). Nevertheless, the experiments show a big advantage of the sequence-based scheme over the random scheme regardless of the presence of PU.

6. Grid-Based QS Rendezvous (gQ-RDV)

In this section we review our previous work [14, 15]. The grid-based quorum rendezvous (gQ-RDV) protocol has been

developed, and to the best of our knowledge, a grid quorum concept has not been utilized previously in the context of channel allocation. Moreover, other quorum-based protocols do not explicitly deal with the heterogeneous spectrum availability, which is a very challenging issue in CRNs. In the protocol each node maps its channels according to the channel quality without any information exchange. It is assumed that each node maintains a list of channels ordered according to their instantaneous quality. The best channel is mapped according to the chosen grid quorum. Hence, nodes that allocate a common best channel, while having the same number of available channels, will always meet thanks to the quorum intersection property. However, if nodes have a different best channel, the method does not guarantee RDV while cycles of the nodes are not aligned. However, the probability of missed RDV is very low with 1.7%. Regarding the second requirement, this approach has $\sim 10\%$ RDV probability on a single channel in a period; that is, nodes meet often on multiple channels in a period. The biggest asset of this algorithm is that it has an asymmetric channel view and capability of dealing with of the channel *heterogeneity* without any information exchange.

7. Mirror-Torus QS Rendezvous Protocol

The gQ-RDV protocol shortly described in the preceding section has a specific feature where the bad-quality channels are less frequented than the good-quality channels. This incurs lesser probability for RDV on these channels. One might argue whether this is an advantage or a drawback. From one hand, it might be better to visit the bad-quality channels (in this paper we follow the nomenclature of quorum system research, where typically a bad channel for a SU means the channel frequently visited by PUs. In the case of wireless communications research the bad channel condition is, of course, related to high interference or low SNR situation. The reader should not confuse these meanings) less frequently since we diminish a possibility of PU disturbance, but on the other hand, the measurements of bad-good quality channels might be not accurate; in other words, we might diminish unnecessary number of RDVs on a particular channel. Therefore, we propose a channel mapping algorithm (MtQS-DSrdv) which aims to guarantee a RDV on all channels. Moreover, each channel is mapped to the same number of slots. Let us focus on the case when a pair of nodes has the same number of available channels in the list (SCHv) and we do not allow to prioritize channels. Based on a spectrum detection algorithm (sensing, database) each CR has a list of spectrum holes that can be used while respecting PU activity. It is assumed that channels are slowly time varying; that is, the system is only slowly dynamic. Here we focus on a distributed rendezvous (RDV) protocol, being one step before an actual medium access.

The MtQS-DSrdv algorithm makes use of frequency hopping sequence. This process requires no mutual knowledge of hopping sequence information and available channels from other CRs. Each CR determines its own hopping sequence, a channel map, from a torus array ($r \times s$) using the torus QS

and DS concepts. Each node has a list of available channels r . The period is composed of the number of slots within which nodes hop from one channel to another, that is, the number of elements in the torus array. Each channel, m_i , has its own index i , where $i = 1 \cdots r$, r stands for the maximum number of available channels. In order to have the same number of attributed slots to each channel map, a torus ($r \times s$) map of each channel must have s slots; hence, for r channels and $[r, s]$ torus, we need $r + \lfloor s/2 \rfloor = s$ elements, that is $s = (r - 1) \times 2 + 1$, for example:

- (i) 2 channels: $[2, 3]$ array, which gives $2 + \lfloor 3/2 \rfloor = 3$ slots;
- (ii) 3 channels: $[3, 5]$ array, which gives $3 + \lfloor 5/2 \rfloor = 5$ slots.

While forming the channel map, two concepts are employed, namely, torus QS and DS. The first one is straightforward, since we simply select a torus quorum, either forward, backward, or mirror tQS, while a head (column) should follow the construction rules of MtQS-DSrdv, and tail can be chosen randomly. Selecting a channel map by following the *DS rule* means that a relaxed DS set must be found of s slots; that is, s elements are selected so that at least one ordered pair (a_i, a_j) exists in (n, s) difference set, where $k = s$ from the DS definition. Since this step is not so easy (due to careful search of possible DSs), here, a sort of structure can be used (presented later in a figure with exemplary channel maps) starting with five available channels.

The column selection of Channel 1 specifies the start point of a map construction. It means that a CR can select its head in s different ways because of s existing columns. In the examples below the head of Channel 1 is selected randomly to be C_1 .

Since the case of $r < 4$ is rather special we elaborate on it first. While having 2 available channels (Map^{2Chs} in Figure 10), the map of Channel 1 is selected in the tQ forward manner, and Channel 2 according to the tQ backward manner. While having three available channels, Channel 1 is selected according to the tQ forward manner as shown in Figure 10 (Map^{3Chs}), and the next two channels are formed from relaxed DSs, that is, DS_{Ch3} : {4, 6, 7, 9, 13} and DS_{Ch2} : {3, 8, 11, 12, 14}. Note that those DSs are exemplary, since others could also be found.

Starting from four available channels MtQS-DSrdv *general* construction rules can be applied:

- (1) Channel 1 selects column (*head*) i randomly;
- (2) Channel 2 selects *head* $i + \lfloor s/2 \rfloor + 1$; for example, with 4 available channels, for Channel 2 the fifth column (C_5) is chosen, since $1 + \lfloor 7/2 \rfloor + 1 = 5$;
- (3) Channel 3 select *head* s ; for example, with four available channels, for Channel 3 the seventh column (C_7) is chosen;
- (4) Channel 4 selects *head* $i + \lfloor s/2 \rfloor + 2$; this rule does not apply with four available channels since the last one (Channel 4) is chosen as a DS;
- (5) The next remaining channels select their map according to *DS rules*, so that there are enough elements

(in appropriate columns) for tails of the first fourth channels.

- (6) When DS maps for channels are found, *tails* of the first four channels are picked up according to forward, backward, or mirror rule.

Note that nodes meet on each channel at least once. In the figure we can see that starting with 4 available channels the mirror tQ is utilized, changing twice the direction (k_i); namely, Channel 3 selects its tails according to the order $\{-1, 1, -1\}$. With five available channels, Channel 2 is constructed using mostly a backward torus manner except of one mirror element, namely, instead of an element from C_4 that from C_8 .

For the sake of understanding and clarity, the column numbers are skipped and the mirror torus extension is visualized in the case of 6 available channels. The circled elements form a DS for Channels 5 and 6. Channels 3 and 4 are constructed according to the mirror tQ. Channel 4 selects its first two tail slots from C_9 (mirror of C_7) and C_{10} (mirror of column C_6), thus in a forward manner. The next tail elements are selected in a backward manner from $C_{3,4,5}$. For the sake of clarity for the Channel 3 map selection we show the selected slots in such a way as they could have been selected in one direction (while wrapping-around), thus in a forward manner. However, in reality the first two tail slots are picked from C_9 (mirror of C_2), and C_{10} (mirror of C_1), in a backward manner, and the next tail elements from $C_{3,4,5}$ in a forward wrap-around manner. Note that Channel 2 must select one mirror element from C_{10} (instead of C_4) in order to find a DS for the last channel (ChR, $r = 6$). The same situation happens with 7 and 8 available channels; Channels 1 and 2 must select some of their elements in a mirror way in order to form a DS for the last channel. This possibility of changing slots of a tQ justifies a flexibility and usability of a mirror tQ selection in our context.

7.1. MtQS-DSrdv Verification. Our objective is to ensure rendezvous both (a) while cycles of nodes are aligned and (b) when the cycles of nodes are not aligned. We define another type of the Rotation Closure Property, which we call it *map-RCP*.

Definition 10 (Map-Rotation Closure Property). For a map $R1$, and $R2$ of a period $\Theta = \{0, \dots, n - 1\}$ and for all $i \in \Theta$, there is defined for all slot_offset, $\exists i : R1_i \cap R2_{(i+\text{slot_offset}) \bmod n} \neq \emptyset$, where slot_offset $\in \Theta$.

The map of channels of one CR node must be checked (whether CRs meet) with each possible cycle shift (slot_offset $\neq 0$). As channel maps are constructed based on either tQS or DS this means automatically that the map-RCP definition is satisfied, because each channel map satisfies the RCP.

7.2. General MtQS-DSrdv Results. Table 1 shows $\min RDV$ and the RDV probability ($RDV_{\text{multi}}^{\text{ch}}$) on the same channel more that once during a period ($\#$ stands for the number of available channels) for the exemplary map selection depicted

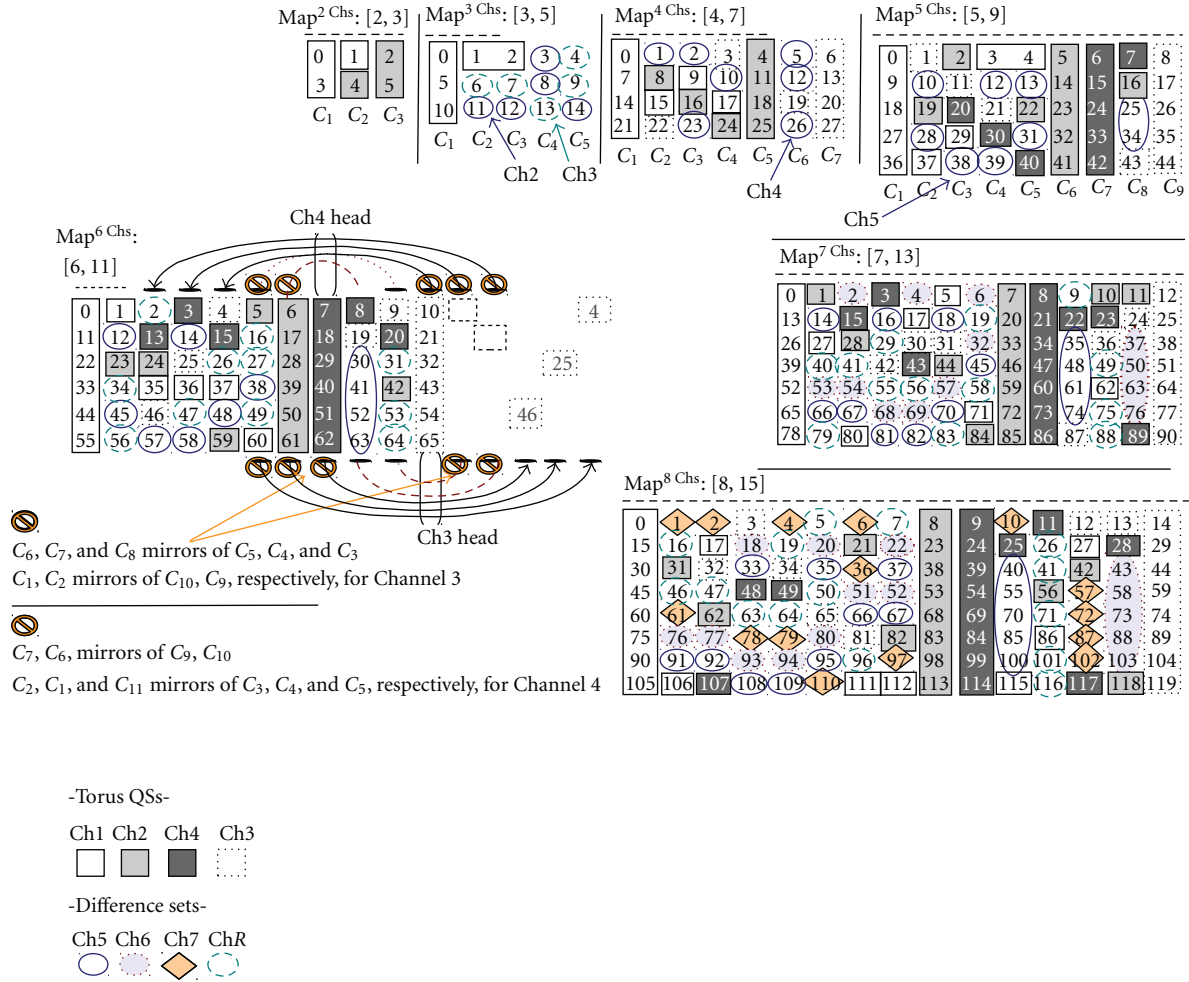


FIGURE 10: MtQS-DSrdv construction exemplary maps (Map^r Chs) for $r = 2 \dots 8$.

on Figure 10. We do not show the results (and maps) for $r > 8$, but the procedure of creating channels follows the rules described in the previous section. $\min\text{RDV}$ means the minimum guaranteed RDV; thus, nodes will always meet on at least once at each channel. It is clear that nodes have more than 30% probability to meet more than once on the same channel in a period ($\text{RDV}_{\text{multi}}^{\text{ch}}$) while $r < 7$, and more than 40% otherwise. Note that $\text{RDV}_{\text{multi}}^{\text{ch}}$ can be improved by selecting better slots for torus-based channels.

7.3. Channel Maps Comparison with Related Work. Figure 11 shows the channel mapping with three available channels according to MtQS-DSrdv protocol (A–C maps), Hsien [65] (D–E maps), A-MOCH [43] (F map), and ACH [63] (G map). In the first map (A) of MtQS-DSrdv nodes B and C use the same channel map ($i = 1$ of Channel 1) but they are shifted relative to node A three and eleventh slots, respectively. We can see that the pair A-B meets five times in a period, A-C four times in A's and C's periods, and B-C has rendezvous three times in their periods.

TABLE 1: Minimum RDV ($\min\text{RDV}$) and RDV probability in multiple slots on the same channel ($\text{RDV}_{\text{Multi}}^{\text{Ch}}$).

#	$\min\text{RDV}$	$\text{RDV}_{\text{Multi}}^{\text{Ch}}$
2	2	33%
3	3	36%
4	4	34%
5	5	34%
6	6	37%
7	7	44%
8	8	45%

In the second map (B) A3 and A5 use different maps, but they are aligned to A, that is, $\text{slot_offset} = 0$. A3 Channel 1 selects third column as its head ($i = 3$) and A5 fifth ($i = 5$). A-A3 pair meets five times in a period, A-A5 four times, and A3-A5 pair five times.

TABLE 2: Comparison according to CH requirements: Yes, Yes/No, No, — (Not elaborated), Yes (all) means RDV on all channels; CHC requirements: (1st) RDV guarantee on at least one channel; (2nd) RDV guarantee on multiple channels; (3rd) TTR small and bounded; (4th) No synchronization/coordination; (5th) No global or neighborhood information; (6th) Symmetric and asymmetric channels.

Scheme/Req.	1st	2nd	3rd	4th	5th	6th	P_{CHs}^{IncSw}	Simplicity
[6]: M-QCH (09)	Yes	Yes	Yes	No	Yes	No	Yes	No
[43]: A-MOCH (11)	Yes	Yes (all)	Yes	Yes	Yes	No	Yes	Yes
[63]: optimal ACH (11)	Yes	Yes (all)	Yes	Yes	Yes	No	Yes/No	Yes
[7]: DSMMAC (11)	Yes	Yes	—	Yes	Yes	No	Yes/No	No
[5]: QRCH (10)	Yes	No	Yes	—	Yes	No	Yes	Yes/No
[60]: CQM (09)	Yes/No	—	—	No	No	No	Yes	No
[58]: RW (10)	Yes	—	Yes/No	Yes	No	Yes	Yes	No
[41]: MC (10)	Yes/No	—	Yes/No	Yes	No	Yes/No	Yes	No
[59]: JS (11)	Yes	—	Yes/No	Yes	Yes	Yes	Yes	No
[65]: Hsien (11)	Yes	Yes/No	Yes	Yes	Yes	No	Yes/No	No
[14]: gQ-RDV:cA/nCA (11)	Yes/No	Yes/No	Yes	Yes	Yes	Yes	Yes/No	Yes
[This work] MtQS-DSrdv (12)	Yes	Yes (all)	Yes	Yes	Yes	—	Yes/No	Yes/No

In the third map (C) A3 and A5 are additionally shifted *one* slot and *seventh* slots, respectively. The pairs A-A3 and A-A5 have *five* times RDV in a period, and A3–A5 *three* times in a period.

The maps D and E show the sequence-based protocol implemented by Hsien et al. [65]. In the map D node A uses sequence $L_1 = \{1, 1, 1, 2, 3\}$, node B sequence $L_2 = \{2, 2, 1, 2, 3, 3, 1\}$, and node C $L_3 = \{3, 3, 3, 2, 3, 1, 1, 2\}$. Since nodes use different sequences (with different lengths (K)) they are able to meet on each channel within a bounded time in the shown cycles (note, there was no proof whether these sequences comply with the RCP). If we analyze RDV in terms of the amount of RDVs in their cycles ($|K_A| = 5$, $|K_B| = 7$, and $|K_C| = 8$) they do not always meet in each cycle; for example, A-B pair has one RDV in the second B's cycle (and none in third A's).

Map E shows the case while nodes select the same map, but they are rotated (A4 and C4 are shifted 4 slots, and B3 3 slots). We can see easily that nodes meet only once on one channel in a period, when the TTR = $K_{node} - 1$ slots.

In the F map A-MOCH [43], the channel allocation is shown, for the transmitter (Tr) using the map $\{2, 1, 3, 1, 3, 2, 3, 2, 1\}$, the receiver (Rc) has the map $\{3, 1, 2, 3, 1, 2, 3, 1, 2\}$, and the receiver is shifted by one slot forward (Rc1 in the figure). Nodes always meet three times in a period of 9 slots.

The map G of ACH [63] shows the same performance, while the transmitter (T) has the map $\{3, 1, 2, 3, 1, 2, 3, 1, 2\}$, the receiver (R) is mapped to $\{3, 2, 1, 1, 3, 3, 2, 1, 2\}$, and the transmitter is shifted one slot backwards ($T - 1$).

In A-MOCH, CRs always meet three times (once on one channel) in a period. Notice that a transmitter and receiver in A-MOCH, and a transmitter in ACH, switch constantly the channel, which as a consequence diminish their chance to meet when CRs are asynchronous.

Note that the MtQS-DSrdv protocol has at least r two neighboring slots; for example, in Figure 11 the number of available channels is 3, where the map has two neighboring slots on each channel, and sometimes there are three

neighboring slots. Thanks to multiple neighboring slots the chance for a RDV is significantly increased while CRs are asynchronous in comparison with the A-MOCH and ACH schemes.

8. Comparison of Selected Protocols according to CHC Requirements

Many of the proposed protocols for multichannel networks or CRNs focus on merits of their approaches such as no need of a (dedicated) control channel (phase) for signaling, or no need for global clock synchronization, or no need of central controller (access point, cluster-head, etc.). Protocol performance is often compared against random rendezvous algorithm or other typical FH or slotted schemes. We consider these assumptions as being obvious and we pertain only to those algorithms which satisfy at least partly the channel hopping cognitive requirements as defined in Section 4.1. The reader should note that we evaluate all approaches from the perspective of CRNs. However, some of the protocols are not designed for CRNs, but for generic multichannel networks. Thus, our evaluation is not meant to be general, and relevant negative comments do not imply for any protocol that they are not interesting for some other use cases. Table 2 compares a number of algorithms in terms of the six CHC requirements defined in Section 4.1. We chose only those schemes which use either an elaborated algorithm for a channel hopping sequences selection, or FH schemes specifically designed for cognitive radio networks.

We consider M-QCH from [6], but do not include the asynchronous approach (A-QCH) because of its channel selection limitation. It is assumed that m channels are selected *randomly* from N channel (maximum number of channels) in order to construct a CH sequence according to the proposed algorithms. If PU signals are detected in one or more of distinct channels, then those channels are *simply* replaced with other channels. In both works the 6th requirement is not taken into account. Due to the lack

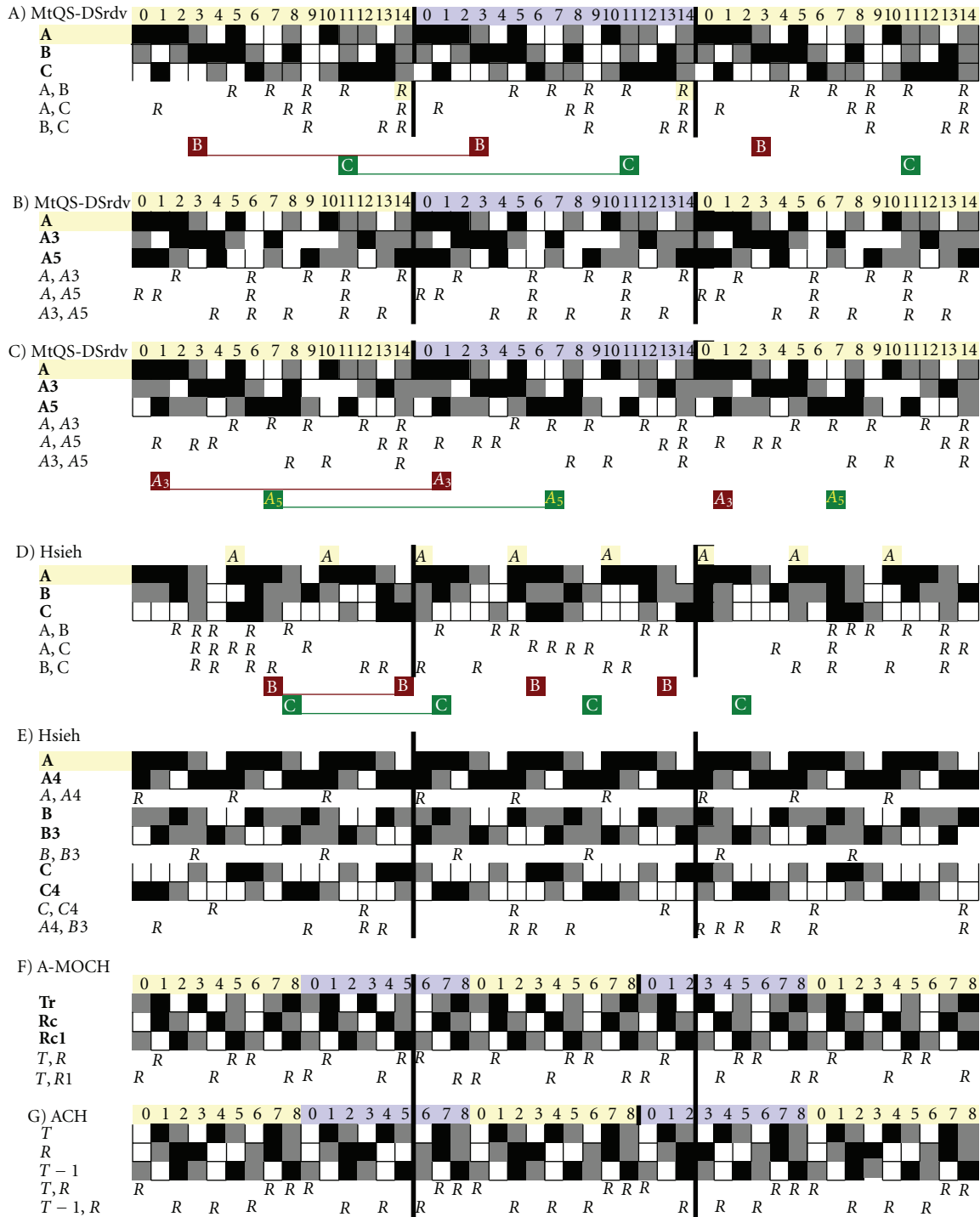


FIGURE 11: Three available channels: MtQS-DSrdv, Hsien et al. [65], A-MOCH [43], and ACH [63] channel mappings; black square stands for Channel 1, gray square for Channel 2, and white square for Channel 3; R stands for rendezvous.

of the heterogeneous spectrum availability consideration, there is a possibility of an increased frequency of channel switching (P_{CHS}^{IncSw}). A-MOCH [43] complies with the first five requirements, but receivers suffer from the maximal frequency of channel switching, and the heterogeneous spectrum availability is not investigated. The optimal ACH

[63] has a similar approach; however, in this scheme receivers switch channels less frequently thanks to better use of the underlying sequence array. DSMMAC [7] might also suffer from the lack of the asymmetric channels' consideration and possible shortcomings related to the use of the same hopping sequence.

In CQM [60] the 1st CHC requirement is yes/no, since it might happen that nodes will not meet because of adopting different default channels. RDV on two or more channels is not elaborated. Moreover, nodes must be synchronized (e.g., by means of GPS devices) and nodes need to know IDs of their 1-hop neighbors. All nodes have the same channel view with no heterogeneity consideration. The protocol aims to share the traffic load evenly by all available channels; however, this is not advantageous in CRNs where some channels can be more frequent occupied than others by PUs.

RW [58] and JS [59] have yes/no regarding the 3rd requirement, since although they guarantee the bounded (E)(M)TTR; it is not a small value. In both papers these approaches are compared with MC algorithms from [41] (having even a larger value(s) of (E)(M)TTR and that one is compared only with a random rendezvous approach), but not with quorum-based ones. All three algorithms do not comply with the 2nd requirement which is essential in CRNs. The Modular Clock (MC) algorithm [41] encounters problems with rendezvous (1st CHC = yes/no) if the selected rates of two users are the same. Its modified version has a similar problem if the selected prime number of two users is the same. The study investigates the possibility of a different channel view; however, the problem is not completely solved.

Hsien et al. [65] sequenced-based protocol guarantees RDV on at least one channel independently of a chosen sequence (different or similar). If a pair of nodes selects a different sequence, then nodes can meet on all channels. However, if the same sequence is used by a pair of nodes then there is only one RDV guaranteed and no more. The protocol is partially based on QS: one channel in each sequence is determined based on QS, and the remaining channels are assigned to the remaining positions in the sequence. Therefore, the creation of sequences seems to be not simple to find in order to ensure RDV on all channels.

The gQ-RDV [14] does not guarantee nonzero probability of RDV while cycles of the nodes are not aligned (nCA case in the table); however, the probability of missed RDV is very low, 1.7%. The approach has ~10% probability to have RDV on a single channel, hence yes/no. The design of this approach supports not only asymmetric channels, but also the channel heterogeneity.

The MtQS-DSrdv protocol always guarantees RDV on all channels in a period. Moreover, multiple RDV probability on the same channel in a period is more than 30%. The first four channel selections are based on the forward, backward, or mirror torus QS, which tremendously simplifies the slot mapping. Other maps are based on the DS concept, which diminishes the simplicity of the algorithm.

Concluding this comparison we can easily see that in some of the schemes some of the requirements are not investigated. None of the approaches satisfies all requirements; each of them is somehow limited or not completely applicable to CRNs. Hence, it is clear that none of those protocols is perfect and there is need for further research. One important conclusion from this review is that different aspects and cross-level work must be taken into account while designing RDV protocols. One could also argue that RDV protocol should be developed together with an

underlying MAC protocol, since the fact that nodes had a chance to meet in a channel does not mean that they can actually successfully communicate.

9. Conclusion

In this paper we have studied quorum systems, originally and widely used in the scope of operating systems. We have reviewed them in the context of cognitive radio networks. We also show that the CCC or a strict synchronization can be omitted by the use of other techniques, still guaranteeing rendezvous among nodes. However, we note that while addressing the RDV problem, different requirements must be satisfied in order to comply with the cognitive radio environment. Finally, as an own contribution we also proposed a novel distributed RDV protocol, which guarantees rendezvous on all channels in a period with a high probability to have multiple RDVs on the same channel in a period.

Acknowledgments

The authors thank the financial support from Deutsche Forschungsgemeinschaft and RWTH Aachen University through UMIC-Research Centre. This work has benefited from discussions with participants of European Union funded ACROPOLIS Network of Excellence.

References

- [1] M. Singhal and N. G. Shivaratri, *Advanced Concepts in Operating Systems*, TMH, McGraw-Hill, New York, NY, USA, 1994.
- [2] A. Kumar, "Hierarchical quorum consensus: a new algorithm for managing replicated data," *Journal IEEE Transaction on Computers*, vol. 40, no. 9, pp. 996–1004, 1991.
- [3] H. Garcia-Molina and D. Barbara, "How to assign votes in a distributed systems," *Journal of the ACM*, vol. 32, no. 4, pp. 841–860, 1985.
- [4] B. Zhang, K. Han, B. Ravindran, and E. D. Jensen, "RTQG: Real-time quorum-based gossip protocol for unreliable networks," in *Proceedings of the 3rd International Conference on Availability, Security, and Reliability (ARES '08)*, pp. 564–571, Barcelona, Spain, March 2008.
- [5] E. K. Lee, S. Y. Oh, and M. Gerla, "Randomized channel hopping scheme for anti-jamming communication," in *Proceedings of the IFIP Wireless Days (WD '10)*, Venice, Italy, October 2010.
- [6] K. Bian, J. M. Park, and R. Chen, "A quorum-based framework for establishing control channels in dynamic spectrum access networks," in *15th Annual ACM International Conference on Mobile Computing and Networking, MobiCom 2009*, pp. 25–36, chn, September 2009.
- [7] F. Hou, L. X. Cai, X. Shen, and J. Huang, "Asynchronous multichannel MAC design with difference-set-based hopping sequences," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 4, pp. 1728–1739, 2011.
- [8] K. Bian and J. Park, "Addressing the hidden incumbent problem in 802.22 networks," in *Proceedings of the Technical Conference and Product Exposition, SDR Forum*, 2009.

- [9] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.
- [10] S. Romaszko, D. Denkovski, V. Pavlovska, and L. Gavrilovska, "Asynchronous rendezvous protocol for cognitive radio ad hoc networks," in *Proceedings of the EAI International Conference on Ad Hoc Networks (ADHOCNETS '12)*, Paris, France, October 2012.
- [11] S. Lang and L. Mao, "A torus quorum protocol for distributed mutual exclusion," in *Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS '98)*, Taiwan, December 1998.
- [12] S. Romaszko, "Making a blind date the guaranteed rendezvous in cognitive radio ad hoc networks," in *Proceedings of the 18th European Wireless Conference (EW '12)*, Poznan, Poland, April 2012.
- [13] S. Romaszko and P. Mähönen, "Torus quorum system and difference set-based rendezvous in cognitive radio ad hoc networks," in *Proceedings of the International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom '12)*, Stockholm, Sweden, June 2012.
- [14] S. Romaszko and P. Mähönen, "Grid-based channel mapping in cognitive radio ad hoc networks," in *Proceedings of the 22nd Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '11)*, Ottawa, Canada, September 2011.
- [15] S. Romaszko and P. Mähönen, "Quorum-based channel allocation with asymmetric channel view in cognitive radio networks," in *Proceedings of the MSWiM Poster and 6th ACM PM2HW2N Workshop*, Miami, Fla, USA, October–November 2011.
- [16] W. S. Luk and T. T. Wong, "Two new quorum based algorithms for distributed mutual exclusion," in *Proceedings of the 17th International Conference on Distributed Computing Systems*, pp. 100–106, May 1997.
- [17] J. R. Jiang, Y. C. Tseng, C. S. Hsu, and T. H. Lai, "Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks," in *Proceedings of the IEEE International Conference on Parallel Processing (ICPP '03)*, Taipei, Taiwan, October 2003.
- [18] J. R. Jiang, Y. C. Tseng, C. S. Hsu, and T. H. Lai, "Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks," *Mobile Networks and Applications*, vol. 10, no. 1, pp. 169–181, 2005.
- [19] M. Maekawa, "A \sqrt{N} algorithm for mutual exclusion in decentralized systems," *ACM Transactions on Computer Systems*, vol. 3, no. 2, pp. 145–159, 1985.
- [20] Y. C. Tseng, C. S. Hsu, and T. Y. Hsieh, "Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '02)*, pp. 200–209, New York, NY, USA, June 2002.
- [21] C.-M. Chao and X. H. Lin, "A fuzzy control quorum-based energy conserving protocol for IEEE 802.11 ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 2180–2185, Hong Kong, March 2007.
- [22] Z. T. Chou, Y. H. Lin, and R. H. Jan, "Optimal fully adaptive power management protocols for asynchronous multi-hop ad hoc wireless networks," in *Proceedings of the 11th IEEE Singapore International Conference on Communication Systems (ICCS '08)*, pp. 569–573, Guangzhou, China, November 2008.
- [23] C. M. Chao, J. P. Sheu, and I. C. Chou, "An adaptive quorum-based energy conserving protocol for IEEE 802.11 ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 5, pp. 560–570, 2006.
- [24] J. M. Hall, *Combinatorial Theory chapter 1*, John Wiley and Sons, New York, NY, USA, 1986.
- [25] Y. C. Kuo, "Quorum-based power-saving multicast protocols in the asynchronous ad hoc network," *Computer Networks*, vol. 54, no. 11, pp. 1911–1922, 2010.
- [26] Y. Manabe, R. Baldoni, M. Raynal, and S. Aoyagi, "K-Arbitrator: A safe and general scheme for h-out-of-k mutual exclusion," *Theoretical Computer Science*, vol. 193, no. 1-2, pp. 97–112, 1998.
- [27] C. H. Wu, J. H. Hong, and C. W. Wu, "Rsa cryptosystem design based on the chinese remainder theorem," in *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC '01)*, Yokohama, Japan, January–February 2001.
- [28] R. H. Thomas, "A majority consensus approach to concurrency control," *ACM Transactions on Database Systems*, vol. 4, no. 2, pp. 180–209, 1979.
- [29] D. Agrawal and A. Abadi, "An efficient and fault-tolerant solution for distributed mutual exclusion," *Journal of ACM Transactions on Database Systems (TODS)*, vol. 9, no. 1, pp. 1–20, 1991.
- [30] S. H. Wu, M. S. Chen, and C. M. Chen, "Fully adaptive power saving protocols for ad hoc networks using the Hyper Quorum System," in *Proceedings of the 28th International Conference on Distributed Computing Systems (ICDCS '08)*, pp. 785–792, Beijing, China, July 2008.
- [31] R. Friedman, G. Kliot, and C. Avin, "Probabilistic quorum systems in wireless ad hoc networks," *ACM Transactions on Computer Systems*, vol. 28, no. 3, p. 7, 2010.
- [32] S. Lai, *Heterogeneous quorum-based wakeup scheduling for duty-cycled wireless sensor networks [Ph.D. thesis]*, Virginia Polytechnic Institute and State University, Blacksburg, Va, United States, 2009.
- [33] S. Lai, B. Zhang, B. Ravindran, and H. Cho, "CQS-Pair: cyclic quorum system pair for wakeup scheduling in wireless sensor networks," *IEEE Transaction on Computers*, vol. 54, pp. 295–310, 2008.
- [34] S. H. Wu, M. S. Chen, and C. M. Chen, "An asymmetric quorum-based power saving protocol for clustered ad hoc networks," in *Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS '07)*, Ontario, Canada, June 2007.
- [35] S. H. Wu, C. M. Chen, and M. S. Chen, "AAA: asynchronous, adaptive, and asymmetric power management for mobile ad hoc networks," in *Proceedings of the 28th Conference on Computer Communications (IEEE INFOCOM '09)*, pp. 2541–2545, Rio de Janeiro, Brazil, April 2009.
- [36] B. J. Choi and X. Shen, "Adaptive asynchronous clock based power saving protocols for delay tolerant networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '09)*, Honolulu, Hawaii, USA, December 2009.
- [37] H. Memarzadeh, M. Dehghan, and S. Jabbehdari, "A new quorum-based power saving protocol to maintain QoS for MANETs with burst traffics," in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA '10)*, pp. 674–679, Melbourne, Australia, April 2010.
- [38] Z. T. Chou, "Optimal adaptive power management protocols for asynchronous wireless ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 61–65, Hong Kong, March 2007.

- [39] T. J. Ross, *Fuzzy Logic with Engineering Applications*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2005.
- [40] S. D. Lang and L. J. Mao, "Comparison of two torus-based k-coterics," in *Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS '98)*, pp. 300–306, Tainan, Taiwan, December 1998.
- [41] 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, 2010.
- [42] B. F. Lo, "A survey of common control channel design in cognitive radio networks," *Physical Communication*, vol. 4, no. 1, pp. 26–39, 2011.
- [43] 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.
- [44] W. Hu, D. Willkomm, M. Abusubaih et al., "Dynamic frequency hopping communities for efficient IEEE 802.22 operation," *IEEE Communications Magazine*, vol. 45, no. 5, pp. 80–87, 2007.
- [45] H. Su and X. Zhang, "Channel-hopping based single transceiver MAC for cognitive radio networks," in *Proceedings of the 42nd Annual Conference on Information Sciences and Systems (CISS '08)*, pp. 197–202, Princeton, NJ, USA, March 2008.
- [46] L. Jiao and F. Y. Li, "A single radio based channel datarate-aware parallel rendezvous MAC protocol for cognitive radio networks," in *Proceedings of the IEEE 34th Conference on Local Computer Networks (LCN '09)*, pp. 392–399, Oslo, Norway, October 2009.
- [47] S. Geirhofer, J. Z. Sun, L. Tong, and B. M. Sadler, "Cognitive frequency hopping based on interference prediction: theory and experimental results," in *Proceedings of the ACM SIGMOBILE Mobile Computing and Communications*, vol. 13, pp. 49–61, 2009.
- [48] E. D. D. S. Calvanese and M. G. D. Benedetto, "A survey on MAC protocols for cognitive radio networks," *IEEE Communications Surveys and Tutorials*, vol. 99, pp. 1–24, 2011.
- [49] B. F. Lo, I. F. Akyildiz, and A. M. Al-Dhelaan, "Efficient recovery control channel design in cognitive radio ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 9, pp. 4513–4526, 2010.
- [50] L. A. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," in *Proceedings of the 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '08)*, pp. 440–446, Ann Arbor, Mich, USA, October 2008.
- [51] P. Pawelczak, S. Pollin, H. S. W. So et al., "State of the art in opportunistic spectrum access medium access control design," in *Proceedings of the 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom '08)*, Singapore, May 2008.
- [52] IEEE, "P802.22d4.0 draft standard for wireless regional area networks part 22: cognitive wireless ran medium access control (MAC) and physical layer (PHY) specifications: policies and procedures for operation in the tv bands," *IEEE*, vol. 45, p. 372, 2008.
- [53] D. Willkomm, W. Hu, D. Hollos, J. Gross, and A. Wolisz, "On centralized and distributed frequency assignment in cognitive radio based frequency hopping cellular networks," in *Proceedings of the 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (CogART '10)*, November 2010.
- [54] M. D. Silvius, F. Ge, A. Young, A. B. MacKenzie, and C. W. Bostian, "Smart radio: spectrum access for first responders," in *Wireless Sensing and Processing III*, usa, March 2008.
- [55] C. Cormio and K. R. Chowdhury, "An adaptive multiple rendezvous control channel for cognitive radio wireless ad hoc networks," in *Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops '10)*, pp. 346–351, Mannheim, Germany, April 2010.
- [56] M. D. Silvius, A. B. MacKenzie, and C. W. Bostian, "Rendezvous MAC protocols for use in cognitive radio networks," in *Proceedings of the IEEE Military Communications Conference (MILCOM '09)*, October 2009.
- [57] C. F. Shih, T. Y. Wu, and W. Liao, "DH-MAC: a dynamic channel hopping MAC protocol for cognitive radio networks," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, Seattle, Wash, USA, May 2010.
- [58] H. Liu, Z. Lin, X. Chu, and Y. W. Leung, "Ring-walk based channel-hopping algorithms with guaranteed rendezvous for cognitive radio networks," in *Proceedings of the IEEE/ACM International Conference on Green Computing and Communications (GreenCom '10), International Conference on Cyber, Physical and Social Computing (CPSCom '10)*, pp. 755–760, Dalian, China, December 2010.
- [59] 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 the IEEE INFOCOM 2011*, pp. 2444–2452, Shanghai, China, April 2011.
- [60] C. M. Chao, H. C. Tsai, and K. J. Huang, "A new channel hopping MAC protocol for mobile ad hoc networks," in *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP '09)*, Nanjing, China, November 2009.
- [61] C. M. Chao and Y. Z. Wang, "A multiple rendezvous multichannel MAC protocol for underwater sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '10)*, Sydney, Australia, April 2010.
- [62] 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 the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, pp. 216–230, Philadelphia, Pa, USA, October 2004.
- [63] K. Bian and J. M. Park, "Asynchronous channel hopping for establishing rendezvous in cognitive radio networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '11), Mini-Conference*, pp. 236–240, Shanghai, China, April 2011.
- [64] I. F. Akyildiz, W. Y. Lee, and K. R. Chowdhury, "CRAHNs: cognitive radio ad hoc networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 810–836, 2009.
- [65] Y. S. Hsieh, C. W. Lien, and C. T. Chou, "A multi-channel testbed for dynamic spectrum access (DSA) networks," in *Proceedings of the 6th ACM workshop on Wireless multimedia networking and computing (WMuNeP '11)*, 2011.
- [66] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '03)*, pp. 35–45, Annapolis, Md, USA, June 2003.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

