Torus Quorum System and Difference Set-based Rendezvous in Cognitive Radio Ad Hoc Networks

Sylwia Romaszko, Petri Mähönen
Institute for Networked Systems, RWTH Aachen University, Kackerstrasse 9, 52072 Aachen, Germany
Email: {sar, pma}@inets.rwth-aachen.de

***Abstract*—In Cognitive Radio Ad Hoc Networks (CRANs) a frequently changing environment and Primary Users channel occupancy result in a need of the searching for a common control channel by Secondary Users in order to be able to initiate a communication. A rendezvous (RDV) is therefore a new challenge, not encountered in single channel networks. Hence, methods guaranteeing that all nodes meet periodically in reasonable periods of time should be advocated, i.e., instead of using pseudo-random solutions, systematic approaches should be considered, e.g, *quorum systems* (QSs) popular especially in power saving protocols. In this study, we propose a novel rendezvous protocol (tQS-DSrdv) which allows to form channel maps based on torus Quorum System (tQS) and difference set (DS) concepts. Thanks to the use of both concepts we are able to guarantee RDV on all r channels when a number of channels is small ($r \leq 4$) and on $r-1$ channels otherwise.**

I. Introduction

A Cognitive Radio Network (CRN) is a network where Primary Users (PUs) have the exclusive right to certain spectrum bands, whereas Secondary Users (SUs) (Cognitive Radio (CR) users), have only an opportunistic spectrum access. Therefore, they can temporarily use a licensed band (overlay spectrum sharing [1]), 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 band because of the PU activity there. A common control channel (CCC) in CRNs supports the transmission coordination exchange and cooperation between the CR users. It is aimed to facilitate neighbor discovery, control signaling, exchange of local measurements etc. However, such CCC assumption in CRNs may not always be feasible and therefore a meeting point (rendezvous) between nodes becomes a big challenge. Avoiding a global synchronization among users, the use of a (dedicated) CCC, or knowledge of users' channel selection strategy, but also the decrease of PUs' disturbance by SUs activity should be considered in such networks. One of well know techniques to decrease the probability of disturbance to PUs is a frequency hopping technique thanks to the frequent switching of the occupied channels.

In this study, we focus on a *distributed* rendezvous protocol for CRANs, also making use of frequency hopping technique. Before going into details of the protocol, we formulate first the rendezvous problem with regard to channel switching. A RDV problem in multichannel networks refers to the ability of two or more CRs to meet each other in the same channel as depicted (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 (both CRs are within its range). In the time period of 15 slots, CR1 user meets CR2 user on Channel 3 in slot 5 and on Channel 4 in slot 12. Users never meet on Channel 2 and 1.

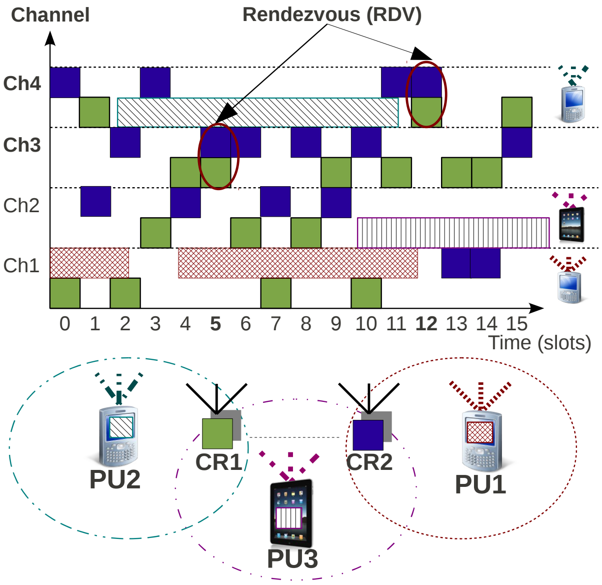

Fig. 1. Blind date, rendezvous problem in CRNs

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. In order to deal with the aforementioned RDV problem, we based our solution on the torus quorum system (tQS) concept. Thanks to the use of the quorum system (QS) and also the difference set (DS) concept we are able to guarantee meeting on $r-1$ channels (r stands for the the number of available channels) while $r > 5$, and on all channels otherwise. The QS intersection property guarantees that nodes will meet (intersect) when selecting quorums from the same quorum system. Quorums, satisfying the *Rotation Closure Property*, guarantee the intersection even if the cycles of the nodes are not aligned and, therefore, can be used in asynchronous protocols (e.g., already used in asynchronous power saving protocols). The *torus QS* [2], represents an intuitive and simple method to employ, by selecting a column of an $r \times s$ rectangular

CROWNCOM 2012, June 18-20, Stockholm, Sweden

DOI 10.4108/icst.crowncom.2012.248284

array (r is the number of rows and s is the number of columns) and $\lfloor \frac{s}{2} \rfloor$ elements from other following columns. Other QS methods are much more complex, e.g., cyclic QS[1] proposed in [4]. The QS concept itself was already presented [5] in the scope of operating systems, in the distributed mutual exclusion, but also widely used for a consistent data replication, agreement problem or dissemination of information. In the last decade, the QS properties usage has been extended to other applications, especially power-saving (PS) protocols ([6], [7] etc.), but also node localisation, gossip protocols for dynamic and reliably discovering eligible nodes. Lately QSs are adopted in order to overcome a rendezvous problem in multichannel networks ([8], [9], [10], [11]).

The objective of this paper is twofold. First, we elaborate on the quorums systems (QS) focusing on a torus QS and its properties. We show that we can form a quorum not only in a standard way [2], but also using an innovative concept, which we call the mirror method of forming a torus Q, being much more flexible while searching quorums in a torus array. Second, frequency hopping sequences are created by selecting either a torus QS or Difference Set which indicates the time slots per period for each channel.

Section II presents the related work. Then we describe the QS and DS concepts with all relevant definitions and their properties. Afterwards we explain the use of torus QS and DS while constructing channel maps. Section IV describes a new torus QS rendezvous (tQS-DSrdv) protocol. In Section V we show tQS-DSrdv verification. Section VI concludes this study.

II. Related work

The habitually assumed support in a cognitive spectrum access is a strict coordination or some degree of synchronization between nodes which reduces the RDV problem to the minimum. Either a TDMA like access scheme is used or Frequency Hopping approach but also assuming that nodes can synchronize/coordinate easily in order to exchange (new) hopping sequences ([12], [13], [14], [15] etc.). In a more recent study [16] it is argued that sequence-based design has little adaptation to PU activity, giving as example an older reference [17], and a scheme more robust to PU activity also from the older study [18]. However, there are many different sequence-based approaches, designed recently (where [17] reference is a more fundamental older one), being more or less adaptable to the PU activity, where the adaptability to the PU activity depends on the design and objective of an algorithm.

We divide the most recent related work into three branches, first, non-quorum based solutions representing blind or pseudo-random RDV techniques ([19], [20], [21] or more sophisticated ones [22], [23]). To the second branch belong protocols proposed for a multi-channel Medium Access Control (MAC) handling multi-rendezvous [24] (i.e., multiple transmissions pairs can accomplish handshaking simultaneously), missing receiver problem [25] and medium allocation in a hostile and jamming environment [8]. All these protocols are based on cyclic quorum systems. A-MOCH presented in [26], is based on Latin Square (LS) (transmitter) and Identical-Row Square maps (receiver). This approach guarantees RDV on all channels, but only once in a period on each channel. Moreover, while being in the reception mode, there are no neighboring slots with the same channel, receiving nodes need to switch channels constantly, which is definitely not desirable in CRANs. It might also happen (example in [26]) that a transmitter will select such LS, which also implies switching channels constantly. In other words, in this approach the channel switching time cost of a receiver (and sometimes for transmitters) should be accounted, since the spectrum mobility events are maximal. Here, a related and very important issue is the energy waste imposed by the constant hopping, but also the lack of consideration of the asymmetric channel view possibility due to the use of LS concept.

To the last branch belong the quorum-based protocols proposed for CRNs ([9], [10], [11]). Three different approaches are proposed in [9] out of which two are synchronous approaches under the assumption of global clock synchronization. The protocols focus either on TTR minimization or even distribution of the rendezvous points in terms of time and channels. The third approach is an asynchronous method, however, designed for systems with only two distinct RDV channels, i.e., one selected based on a minimal cyclic QS, the second based on a majority cyclic QS. In [10] all nodes create the same channel hopping sequence, and the only possible difference of this sequence is dependent on an offset (if cycles are not aligned). Moreover, the process of the forming of channel hopping sequences is not easy, namely it is based on the difference (and complementary) sets which must be chosen in very careful manner in order to ensure a high RDV probability. Since a torus QS is a special case of a grid QS, we also address two grid quorum based rendezvous (gQ-RDV) algorithms from our previous work [11]. The gQ-RDV schemes do not guarantee RDV, although the percentage of missed RDV is a very low ($\sim 2\%$). The RDV on two different channels is predominant in these schemes, but RDV on all channels is rather rare (mostly $\sim 10\%$). In contrary to gQ-RDV method, tQS-DSrdv protocol guarantees rendezvous on all r available channels while $r < 5$, and $r - 1$ channels otherwise.

III. Quorum Systems

In this section we present the quorum system (QS) concept focusing especially on a torus QS, since this type is used in the construction of our channel maps. Afterwards, we also elaborate on the Difference Sets concept as it is the second main component to form our channel maps. Since QSs are not commonly used in wireless communications, we present first some relevant definitions in this section. Quorum-based algorithms become popular as the main asset of these algorithms is their resilience to node and network failures. Basically, a quorum is a collection of sets that intersect with each other at least once within a certain period. This property

[1] A cyclic QS is based on the cyclic block design and cyclic DSs in combinatorial theory [3].

makes QS interesting to solve a RDV problem. The usual definition is given in [6]:

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

There are different types of QSs out of which a torus-based QS (tQS) [2] is less frequently used than others, but a noteworthy system. tQS is similar to the grid-based QS [4], but adopting a rectangular array structure (instead of a $n \times n$ grid) called torus, i.e., wrap-around mesh, where the last row (column) is followed by the first row (column) in a wrap-around manner. The height *r* (number of rows, i.e. entire column) and width *s* (number of columns, i.e. entire row) are defined where $n = r \times s$ and $s \geq r \geq 1$.

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

Figure 2 illustrates an example for three nodes, A, B, C, choosing three different torus quorums under $U = \{0, 1, ..., 17\}$ 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 a 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 slot 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).

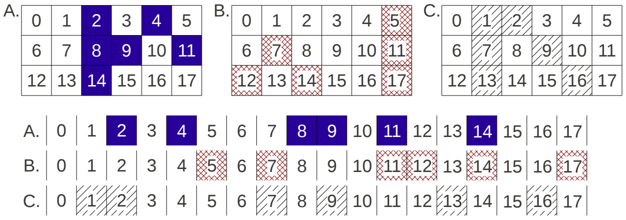

Fig. 2. Three quorums of the standard (*forward*) tQS in a 3x6 torus

The aforementioned definition is a *standard* definition found in the literature [2]. We found that it is also possible to construct tQ in a backward manner [27] (Figure 3). The standard wrap-around torus concept is still followed, but for a tail ($\lfloor \frac{s}{2} \rfloor$ elements) one element is selected out of each of the $\lfloor \frac{s}{2} \rfloor$ preceding columns (c_{j-i}, $i = 1..\lfloor \frac{s}{2} \rfloor$) using begin wrap-around. Nodes A, B, C (Figure 3) intersect twice in slot 4, 11 (A-C), 9, 14 (A-B) and 7, 12 (B-C).

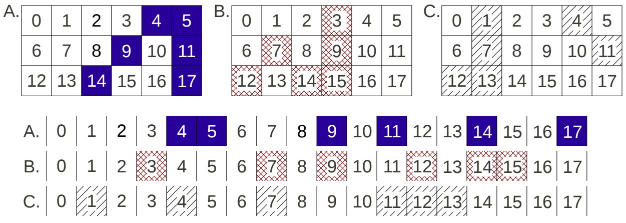

Fig. 3. Three quorums of the *backward* tQS in a 3x6 torus

We can form a torus walking forward and backward, in this case, why not in both directions? A torus Q can be selected in wrap-around manner after all. Following this observation we define the *mirror* torus extension, which allows to select a tail in much more flexible manner:

Definition 3: Mirror Torus Extension : A tail of a torus quorum, $\lfloor \frac{s}{2} \rfloor$ elements, can be selected from any position of column c_{j+k_i*i} (one element from a column), where $k_i \in \{1, -1\}$ and $i = 1..\lfloor \frac{s}{2} \rfloor$, 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 needs to be the same for all quorums of the same torus QS, i.e. the direction of the selection needs to be the same. Figure 4 shows the selection in a mirror way. Node A and B initiate their tail selection in a backward manner, i.e., the first element of node A is selected from column c_5 (mirror of c_1), and the one of node B is selected from column c_4 (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 case of node A, and c_1 in case of node B. The last element of node 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.

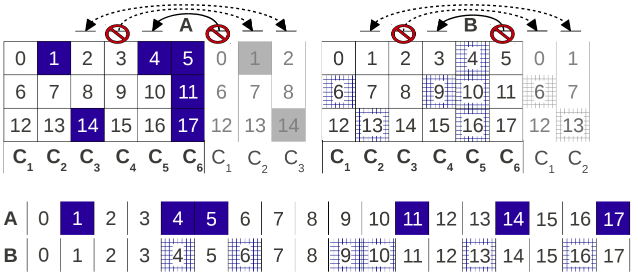

Fig. 4. Mirror Torus QS concept

The QS intersection property is not sufficient when the cycle of nodes are not aligned or nodes are asynchronous. In order to have RDV guarantee in such case, a quorum must satisfy Rotation Closure Property (RCP). A QS which satisfies the RCP, ensures that two asynchronous mobile nodes selecting any two quorums from the same QS have at least one intersection in their quorums [6]. The (forward, backward and mirror) tQS satisfies the RCP:

Definition 4 Rotation Closure Property [6] : For a quorum R in a quorum system Q under an universal set $U = \{0,...,n-\ 1\}$ and $i \in 1,2,...,n-1$, there is defined: $rotate(R,i) = (x\ +\ i)\ mod\ n|x \in R$. A quorum system Q has the *Rotation Closure Property* if and only if $\forall R', R \in Q, R' \cap rotate(R,i) \neq \emptyset$ for all $i \in 1,2,...,n-1$.

For instance, 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 $U = \{0,1,2,3\}$ has no RCP since $\{0,1\} \cap rotate(\{0,3\},3) = \emptyset$.

The Difference Sets (DS) [3] concept is very close to QSs, being actually the basis of the cyclic QS, which is not described in this paper. The DS concept is also widely utilized in PS algorithms. The standard definition of DS as follows:

Definition 5: Difference Set (DS) [3] : A set of k residues $D\ :\ \{a_1,...,a_k\}\ modulo\ n$, $D \in Z_n$, is a (n,k,λ)-difference set if for every $d\ \not\equiv 0(mod\ n)$ there are exactly λ ordered pairs (a_i,a_j), a_i, $a_j \in D$ in such a way that $a_i - a_j \equiv d(mod\ n)$. Z_n denotes the set of nonnegative integers less than n.

For example a set $\{5,6,8,1\}$ of Z_{13} is $(13,4,1)$-difference set because:

$$\begin{array}{ll} 1 \equiv 6-5,\ 2 \equiv 8-6,\ 3 \equiv 8-5,\ 4 \equiv 5-1, & \\ 5 \equiv 6-1,\ 6 \equiv 1-8,\ 7 \equiv 8-1,\ 8 \equiv 1-6, & (mod\ 13) \\ 9 \equiv 1-5,\ 10 \equiv 5-8,\ 11 \equiv 6-8,\ 12 \equiv 5-6. & \end{array}$$

The relaxed Difference Set definition [3], utilized in this paper (and in cyclic QS), says:

Definition 6: Relaxed Difference Set (DS) [4] : A set $D = \{a_1,...,a_k\} \subset Z_n$ is called a relaxed cyclic (n,k)-difference set if for every every $d\ \not\equiv 0(mod\ n)$ there exists at least one ordered pair (a_i,a_j), a_i, $a_j \in D$, such that $a_i - a_j \equiv d(mod\ n)$.

Since tQs also form relaxed DSs we show an example based on the set from Figure 2. Node A has a set $\{2,4,8,9,11,14\}$ of Z_{18}. The set is $(18,6)$-difference set because:

$$\begin{array}{ll} 1 \equiv 9-8,\ 2 \equiv 4-2,\ 3 \equiv 11-8,\ 4 \equiv 8-4,\ 5 \equiv 9-4, & \\ 6 \equiv 8-2,\ 7 \equiv 9-2,\ 8 \equiv 4-14,\ 9 \equiv 11-2, & \\ 10 \equiv 14-4,\ 11 \equiv 2-9,\ 12 \equiv 2-8,\ 13 \equiv 9-14, & (mod\ 18) \\ 14 \equiv 4-8,\ 15 \equiv 11-14,\ 16 \equiv 2-4,\ 17 \equiv 8-9. & \end{array}$$

Note, that from now on, while talking about the Difference Set we refer to the relaxed Difference Set.

IV. Torus QS rendezvous protocol

In this work we focus on a fully *distributed* RDV protocol, thus one step before an actual medium access. We do not elaborate on MAC in this paper. Note that, there are a couple of ways to act after nodes' rendezvous, e.g. nodes stay at the RDV channel and exchange data (with or without the handshake of control packets), or nodes agree to hop synchronously and exchange data. Since the MAC issue is out of the scope we do not go further into details.

tQS-DSrdv algorithm makes use of frequency hopping sequence/s, i.e., we consider that each node hops from one channel to another (Figure 1) in order to communicate with another node. But to do so, first it needs to know how to hop. We do not exchange any knowledge about other nodes hopping sequences as it is usually required in FH schemes. Each node determines its own hopping sequence, a channel map, from a torus array $(r \times s)$ using torus QS and Difference Set concepts. Each node has a list of available channels r (r stands for the maximum number of available channels). In this work we focus on the case when a pair of nodes has the same number of available channels in the list. The period (cycle) is composed of n slots (the number of elements in the torus array) within which nodes hop from one channel to another. Each channel m_i has its own index i known by nodes, where $i = 1..r$. 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 $r + \lfloor \frac{s}{2} \rfloor = s$ elements, i.e., $s = (r-1) \times 2 + 1$, e.g.:

- 3 channels: $[3,5]$ array, which gives $3 + \lfloor \frac{5}{2} \rfloor = 5$ slots
- 4 channels: $[4,7]$ array, which gives $4 + \lfloor \frac{7}{2} \rfloor = 7$ slots

While forming the channel map, two concepts are used as aforementioned, namely torus QS and Difference Sets. The first one is straightforward, since we just select a torus quorum, either forward, backward or mirror tQS, while the head (column) should follow the construction rules of tQS-DSrdv, and tail can be chosen randomly, of course following torus QS rules, which we can relax/adjust as shown below.

Since the case of $r < 4$ is rather special we elaborate on it first. While having 2 available channels, the map of channel 1 is selected in a forward manner of tQ, and channel 2 according to a backward manner. While having 3 available channels, channel 1 is selected according to tQ forward manner as shown in Figure 5 ($Map^{\mathbf{3}\ Chs}$). The next two channels sets are relaxed Difference Sets, e.g., exemplary sets for channel 2 and 3 shown in the figure, DS_{Ch2}: $\{4,6,7,9,13\}$ and DS_{Ch3}: $\{3,8,11,12,14\}$, respectively. Starting from 4 available channels tQS-DSrdv *general* construction rules can be applied:

1) Channel 1 selects *head* i randomly, e.g., in Figure 5 the first column (C_1) is chosen;
2) Channel 2 selects *head* $i + \lfloor \frac{s}{2} \rfloor + 1$; e.g., with 4 available channels, for channel 2 the fifth column (C_5) is chosen, since $1 + \lfloor \frac{7}{2} \rfloor + 1 = 5$;
3) Channel 3 select *head* s; e.g., with 4 available channels, for channel 3 the seventh column (C_7) is chosen;
4) Channel 4 selects *head* $i + \lfloor \frac{s}{2} \rfloor + 2$; this rule does not apply with 4 available channels since the last one (channel 4) is chosen as DS;
5) For the next $r - 4 - 1$ channels DS maps are searched (1) according to the Definition 6., and so that (2) there are enough elements (in appropriate columns) for tails of the first four channels. Here, a sort of structure can be used, circled DSs in Figure 5, starting with 5 channels;
6) The last channel r picks up elements in such a way that (a) there are enough elements for tails of the first four channels; and (b) it has as few as possible missing d from the DS definition, e.g., with 5 available channels the circled set (Figure 5) forms *almost* DS, we say almost

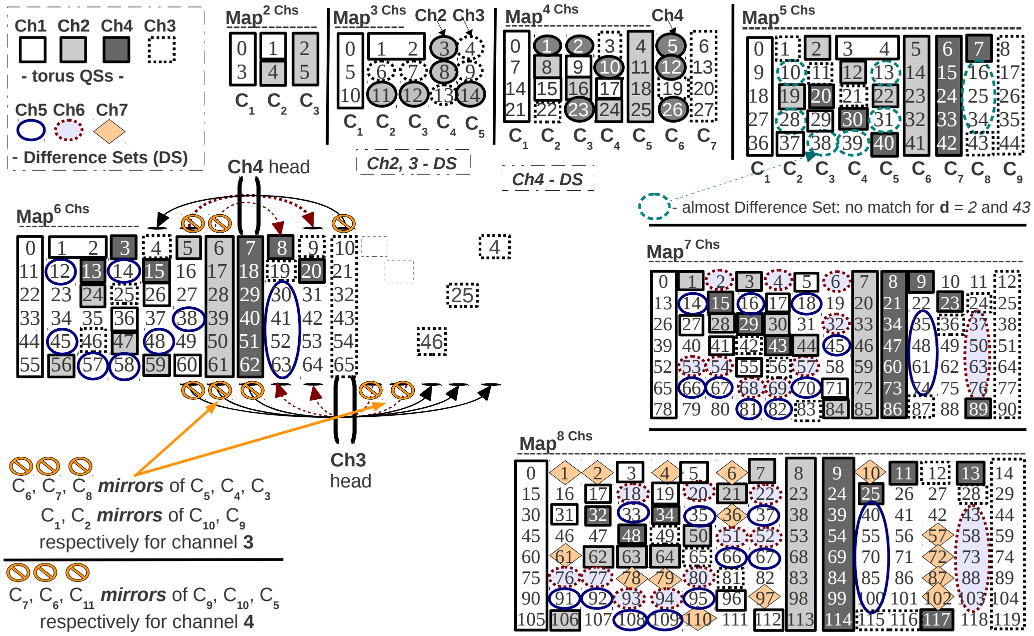
Ch1 Ch2 Ch4 Ch3
- torus QSs -
Ch5 Ch6 Ch7
- Difference Sets (DS)
Map^2 Chs
Map^3 Chs
Ch2 Ch3
Ch2, 3 - DS
Map^4 Chs
Ch4
Ch4 - DS
Map^5 Chs
- almost Difference Set: no match for d = 2 and 43
Ch4 head
Map^6 Chs
Ch3 head
Map^7 Chs
Map^8 Chs
C6, C7, C8 mirrors of C5, C4, C3
C1, C2 mirrors of C10, C9
respectively for channel 3
C7, C6, C11 mirrors of C9, C10, C5
respectively for channel 4

Fig. 5. tQS-DSrdv: $Map^{\mathbf{r}\ Chs}$ stands for a map with r channels; elements without a circled/squares form a set of the last channel with 6, 7, and 8 channels

since for 2 and 43 we cannot find a pair (a_i, a_j) so that $a_i - a_j \equiv d(mod\ n)$. However, knowing that only those two d elements are missing, we also know when nodes will not meet while cycles of nodes A and B are not aligned, i.e., if the cycle of node B is shifted by 2 or 43 slots, then node B will not meet with node A on 5^{th} channel (otherwise nodes will always meet).

7) *Tails* of the first four channels are picked up according to forward, backward or mirror rule.

Note that with 2, 3, and 4 available channels (Figure 5), nodes meet on each channel. In the figure we can see that starting with 4 available channels the mirror tQ is utilized. 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 blue elements form a *DS* for channel 5. Channel 3 and 4 is constructed according to the mirror tQ. Channel 4 selects its first two *tail* slots from column 9 (mirror of column 7) and 10 (mirror of column 6), thus in a forward manner. The next tail elements are selected in a backward manner from column 3, 4, and 5 (note, not all arrows are shown for this channel). For the sake of clarity for the channel 3 map selection we show the selected slots in such as way as they could have been selected in one direction (while wrapping-around the array), thus in a forward manner. However, in reality the first two tail slots are picked from column 9 (mirror of column 2), and 10 (mirror of column 1), thus in a backward manner, and the next tail elements, from columns 3, 4 and 5, thus in a forward wrap-around manner.

V. Verification

All used torus and DS definitions have been implemented in C++ in order to collect individual channel statistics and verify RDV occurrence of a channel in a period, i.e., apart from that nodes meet once on $r-1$ channels, they can also meet two or more times on the same channel during one period. The missed matches of the last channel can be checked by utilizing the DS definition, in such a way we can specify/control in which slot offsets there is no match. Note that the channel maps (sets) constructed based on either tQS or DS satisfy RCP, therefore the map with such channels also satisfies the *map-RCP*:

Definition 7 map-Rotation Closure Property For a map $R1$, and $R2$ of a period $\Theta = \{0, ..., n-1\}$ and $\forall i \in \Theta$, there is defined: $\forall\ slot\_offset, \exists i : R1_i \cap R2_{(i+slot\_offset) mod\ n} \neq \emptyset$, where $slot\_offset \in \Theta$.

In other words, the map of a node must be checked with each possible shifted cycle ($slot\_offset \neq 0$). With a slot offset 0 (cycle alignment case) nodes will always meet on all channels at least once.

Table I shows ${}_{min}$RDV and $Probability_{missed}$ for exemplary maps selection depicted on Figure 5. ${}_{min}$RDV means the minimum guaranteed RDV, thus, nodes will always meet on at least $r-1$ channels. $Probability_{missed}$ stands for the total percentage of possible missed matches on the last channel. Note that $Probability_{missed}$ might be improved by selecting better slots for the last channel.

r	${}_{min}$RDV	$Probability_{missed}$
2, 3, 4	2, 3, 4 resp.	0%
5	4	2 slots: 4.4%
6	5	24%
7	6	29%
8	7	25%

TABLE I
Minimum RDV and Probability of missed RDV on r channels

Figure 6 depicts an intersection distribution of tQS-DSrdv

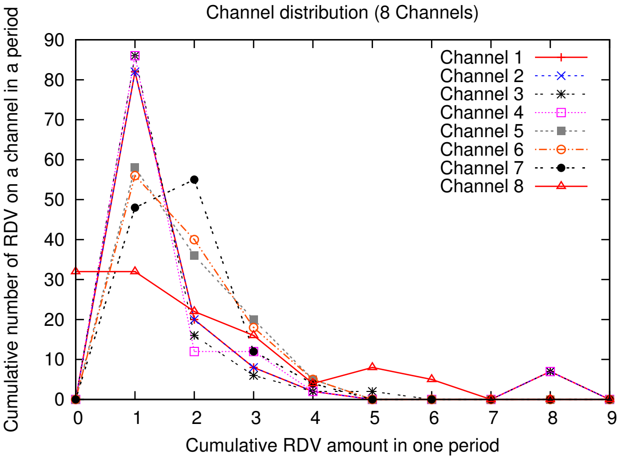

Fig. 6. Channel distribution with 8 available channels including the rotation of the whole cycle

for each possible cycle shift. The RDV distribution on each channel is different. RDV occurrence of a channel only once in a period mostly happens for the first four channels (based on tQ), and less often twice in a period. On the other hand, on DSs channels (5, 6, and 7) nodes meet more often in one period. Notice, that nodes do not meet on channel 8 in 32 possible slots as aforementioned, and in the figure.

VI. Conclusion and Future Work

In this work we developed a novel rendezvous protocol which guarantees RDV on all channels while $r < 5$, and on $r - 1$ channels otherwise. On one hand, the protocol is based on a torus QS, which simplifies tremendously the selection of the channel sets for the first four channels. On the other hand, it is based on the DS concept, which increases complexity somewhat since DSs must be found. Finally, in order to utilize a torus QS concept we have also proposed the mirror torus extension. In our future work we want to obtain RDV always on all channels and we will elaborate on the case with a different number of available channels in a set.

Acknowledgment

We thank the financial support from Deutsche Forschungsgemeinschaft and RWTH Aachen University through UMIC-research centre. This work has benefited from discussions in FP7-ICT NoE ACROPOLIS project.

References

[1] I.F.Akyildiz, W.Y.Lee, M.C.Vuran, and S.Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Elsevier*, vol. 50, pp. 2127–2159, 2006.

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

[3] J. Hall, *Combinatorial Theory chapter 1*. John Wiley and Sons, 1986.

[4] W.S.Luk and T. Wong, "Two new quorum based algorithms for distributed mutual exclusion," in *17th International Conference on Distributed Computing Systems (ICDCS)*, USA, May 1997.

[5] M.Singhal and N.G.Shivaratri, *Advanced concepts in operating systems*. TMH, 1994.

[6] 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 *Parallel Processing*, October 2003.

[7] 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 *11th IEEE Singapore International Conference on Communication Systems (ICCS)*, China, November 2008.

[8] E.K.Lee, S.Y.Oh, and M.Gerla, "Randomized channel hopping scheme for anti-jamming communication," in *Wireless Days*, Venice, Italy, October 2010.

[9] K.Bian, J.M.Park, and R.Chen, "A quorum-based framework for establishing control channels in dynamic spectrum access networks," in *15th annual international conference on Mobile computing and networking (MobiCom)*, China, September 2009.

[10] 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, pp. 1728 – 1739, 2011.

[11] S.Romaszko and P.Mähönen, "Grid-based channel mapping in cognitive radio ad hoc networks," in *PIMRC*, Canada, September 2011.

[12] W.Hung, D.Willkomm, M.Abusubaih, J.Gross, G.Vlantis, M.Gerla, and A.Wolisz, "Dynamic frequency hopping communities for efficient IEEE 802.22 operation." *IEEE Communication Magazine, Special Issue: Cognitive Radios for Dynamic Spectrum Access*, vol. 45, pp. 80–87, 2007.

[13] H.Su and X.Zhang, "Channel-hopping based single transceiver mac for cognitive radio networks," in *42nd Annual Conference on Information Sciences and Systems (CISS)*, USA, March 2008.

[14] L.Jiao and F.Y.Li, "A single radio based channel datarate-aware parallel rendezvous mac protocol for cognitive radio networks," in *Local Computer Networks*, 2009.

[15] S.Geirhofer, J.Z.Sun, L.Tong, and B.M.Sadler, "Cognitive frequency hopping based on interference prediction: Theory and experimental results," *ACM SIGMOBILE Mobile Computing and Communications*, vol. 13, pp. 49–61, 2009.

[16] 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, pp. 4513 – 4526, 2010.

[17] L.A.DaSilva and I.Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," in *3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, USA, October 2008.

[18] P.Pawelczak, S.Pollin, H. So, A.Motamedi, A.Bahai, R. V.Prasad, and R.Hekmat, "State of the art in opportunistic spectrum access medium access control design," in *International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom)*, Singapore, May 2008.

[19] 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 (SPIE)*, 2008.

[20] C.Cormio and K.R.Chowdhury, "An adaptive multiple rendezvous control channel for cognitive radio wireless ad hoc networks," in *Pervasive Computing and Communications Workshops (PERCOM WS)*, Germany, March-April 2010.

[21] N.C.Theis, R.W.Thomas, and L.A.DaSilva, "Rendezvous for cognitive radios," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 216–227, 2010.

[22] H.Liu, Z.Lin, X.Chu, and Y.W.Leung, "Ring-walk based channel-hopping algorithms with guaranteed rendezvous for cognitive radio networks," in *International Workshop on Wireless Sensor, Actuator and Robot Networks (WiSARN2010-FALL), in conjunction with IEEE/ACM CPSCom*, China, December 2010.

[23] Z.Lin, H.Liu, X.Chu, and Y.W.Leung, "Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks," in *IEEE INFOCOM*, China, April 2011.

[24] C.M.Chao, H.C.Tsai, and K.J.Huang, "A new channel hopping mac protocol for mobile ad hoc networks," in *Wireless Communications and Signal Processing (WCSP)*, December 2009.

[25] C.M.Chao and Y.Z.Wang, "A multiple rendezvous multichannel mac protocol for underwater sensor networks," in *WCNC*, Australia, March 2010.

[26] 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, pp. 689–703, 2011.

[27] S. Romaszko, "Making a blind date the guaranteed rendezvous in cognitive radio ad hoc networks," in *18th European Wireless conference*, Poland, April 2012.