Sādhanā Vol. 42, No. 11, November 2017, pp. 1825–1834
DOI 10.1007/s12046-017-0738-1

© Indian Academy of Sciences

Comprehensive asynchronous symmetric rendezvous algorithm in cognitive radio networks

MEENU CHAWLA, AISHWARYA SAGAR ANAND UKEY* and P RESHMA

Department of Computer Science and Engineering, Maulana Azad National Institute of Technology, Bhopal 462003, India
e-mail: chawlam@manit.ac.in; aishwaryaukey@gmail.com; reshma.p@gmail.com

MS received 24 May 2016; revised 23 December 2016; accepted 7 March 2017; published online 10 November 2017

Abstract. Spectrum has become a scant quantity with recent upsurge in the field of wireless communication. Cognitive radio network (CRN) alleviates the overgrowing spectrum scarcity and underutilization problem by adequately sharing the frequency bands between licensed and unlicensed users. CRN allows unlicensed users or secondary users (SUs) to opportunistically utilize the free portion of the spectrum allocated to the licensed users or primary users. The fundamental process in the formation of CRN is the rendezvous process where SUs meet on commonly available channels and establish reliable links for effective communication. Existing rendezvous solutions based on the assumption of a common control channel (CCC) among the SUs are infeasible and less efficient in the dynamic environment of CRNs. Therefore, channel hopping (CH) technique without CCC support, often referred to as blind rendezvous, is usually employed for accomplishing the rendezvous between SUs. This paper presents a comprehensive asynchronous symmetric rendezvous (CASR) algorithm that does not require time synchronization and guarantees rendezvous of SUs in finite time. CASR algorithm exploits the MAC address of SU as the unique identifier (ID) and generates CH sequence based on the dynamic manipulation of ID according to the number of available communication channels. Leveraging the unique ID of each SU, CASR algorithm succeeded in rendezvous guarantee while perpetuating a good time to rendezvous. The efficiency of CASR algorithm is estimated theoretically and verified through various simulation experiments. Simulation results affirm that CASR algorithm performs better in terms of average time-to-rendezvous as compared with existing rendezvous algorithms.

Keywords. Dynamic spectrum access; opportunistic spectrum access; cognitive radio networks; rendezvous process; neighbour discovery; symmetric rendezvous algorithm.

1. Introduction

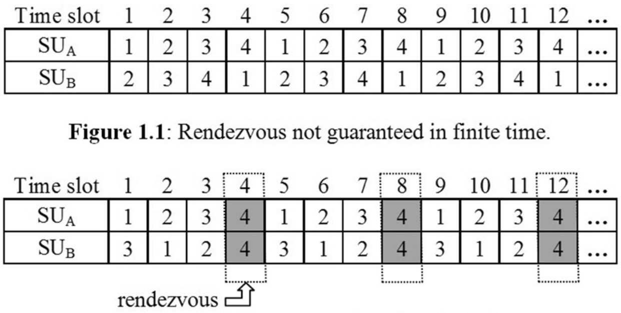
Cognitive radio network (CRN) is an emerging paradigm developed to alleviate the overgrowing scarcity and utilization problem of licensed spectrum. CRN enables unlicensed users or secondary users (SUs) to opportunistically utilize the free portion of spectrum (also referred to as spectrum holes or white spaces [1]) allocated to the licensed users or primary users (PUs). In CRNs, PUs are given higher priority in using the allotted spectrum. SUs can access the licensed spectrum as long as it is not being used by the PUs, and vacate or switch to some other channel on PU reappearance [2]. The key technology behind the CRN is the cognitive radio (CR), which can be defined in terms of its cognitive capability and reconfigurability [1–4]. Leveraging the cognitive capability, CR senses its radio environment and discovers the temporary unutilized portion of the spectrum. Reconfigurability helps CR to dynamically adapt to the varying spectrum environment [5].

The fundamental step during the configuration of CRN is the rendezvous process or neighbour discovery where SUs meet on commonly available channels and establish communication links for information exchange. Traditionally, periodic beaconing is employed in ad hoc networks for the purpose of neighbour discovery where network nodes exchange beacon messages over the pre-defined communication channel and establish communication links between them. However, in CRN, there is no pre-defined communication channel available for propagating beaconing information. Instead, SUs perform spectrum sensing and dynamically figure out the free, available channels through which necessary communication links can be established. Moreover, due to variations in PUs activity and geometric location of SUs, available channels sensed by SUs generally differ and change over time [6]. In such highly dynamic environment of CRN, it becomes extremely difficult to find out a channel that is commonly available to

*For correspondence

all SUs, as available channels change over time. This makes the rendezvous process a non-trivial task. Therefore, it is crucial to bring SUs on common communication channels for exchanging the beacon information [7]. The process of SUs to meet on commonly available channel and to establish the communication links is referred to as rendezvous process [8]. Rendezvous is said to occur between the two SUs whenever both happen to be on the same channel simultaneously for a certain period of time that is sufficient enough to establish a reliable link between them [9].

Several rendezvous algorithms have been presented for achieving the rendezvous among SUs in CRNs. Considering diverse scenarios of CRNs, rendezvous algorithms can be categorized into centralized and distributed, synchronous and asynchronous, and symmetric and asymmetric types [8, 10, 11]. Centralized algorithm requires a central entity and a predefined common control channel (CCC) to control the rendezvous process whereas distributed algorithm is independent of the central server and CCC. Synchronous algorithm restricts SUs to start the rendezvous process simultaneously at the same time, whereas asynchronous algorithm does not require any time synchronization among SUs. Symmetric algorithm assumes the same set of available channels among the SUs, whereas asymmetric algorithm considers different channel availabilities of SUs. Most of the rendezvous algorithms utilize the channel hopping (CH) strategy, which considers a time-slotted system in which time is divided into slots of equal size. SUs hop among channels in a sequence (referred to as CH sequence) with one channel per time slot in order to discover potential neighbours. However, simply hopping on channels with one channel per time slot may not always achieve the rendezvous as SUs may hop to different channels in each time slot. As an example, if the CH sequence generated by two SUs A and B is {1, 2, 3, 4} and {2, 3, 4, 1}, respectively, and the SUs hop as per their CH sequence (illustrated in figure 1.1) then rendezvous will never be achieved. Rendezvous will be achieved only if SUs hop to the same channel at the same time slot as illustrated in figure 1.2. Hence, CH sequence should be devised in such a manner that guarantees rendezvous of SUs in finite time.



Time slot	1	2	3	4	5	6	7	8	9	10	11	12	…
SU_A	1	2	3	4	1	2	3	4	1	2	3	4	…
SU_B	2	3	4	1	2	3	4	1	2	3	4	1	…

Figure 1.1: Rendezvous not guaranteed in finite time.

Time slot	1	2	3	4	5	6	7	8	9	10	11	12	…
SU_A	1	2	3	4	1	2	3	4	1	2	3	4	…
SU_B	3	1	2	4	3	1	2	4	3	1	2	4	…

rendezvous

Figure 1.2: Rendezvous guaranteed in finite time.

Figure 1. Channel hopping rendezvous strategy.

To evaluate the performance of rendezvous algorithm, time-to-rendezvous (TTR) is used as the most prominent metric, which refers to the number of time slots taken to attain rendezvous once all SUs have started the rendezvous process. However, in the asynchronous environment of CRN, SUs are free to start the rendezvous process at any point of time. Hence, maximum TTR (MTTR) and expected TTR (ETTR) are usually employed as major performance measuring metrics. MTTR and ETTR refer to the time taken for rendezvous in the worst case and average case, respectively [12–14].

This paper presents a comprehensive asynchronous symmetric rendezvous (CASR) algorithm that does not require time synchronization among SUs and guarantees the rendezvous in finite time. The presented algorithm utilizes unique IDs of SUs and constructs the CH sequence based on dynamic manipulation of IDs according to the number of available channels. Performance of presented algorithm is evaluated theoretically in terms of ETTR and is verified through several simulation experiments. Simulation results show that the CASR algorithm performs much better as compared with existing state-of-art rendezvous solutions. Rest of the paper is organized as follows. Section 2 introduces some of the CH rendezvous algorithms. Section 3 describes the proposed CASR algorithm in detail. Section 4 includes the performance evaluation of the proposed work. Section 5 covers the simulation results and comparative analysis. Section 6 concludes the paper and highlights the future work.

2. Related work

Earlier CH rendezvous algorithms such as the algorithms proposed by Cormio and Chowdhury [15] and Kondareddy *et al* [16]) employ the random strategy where SUs design the CH sequences among the available channels in a random way. Though random strategy might be able to procure rendezvous, it failed to guarantee the rendezvous in a bounded time since the channel switching performed by the SUs is quite unpredictable.

Yang *et al* [17] presented the deterministic rendezvous sequence (DRSEQ) algorithm, which guarantees the rendezvous in finite time under the symmetric model. The CH sequence generated by DRSEQ exhibits the pattern $1, 2, \ldots, N, e, N, N-1, \ldots, 1$ where e denotes an empty slot. DRSEQ has MTTR of $2N+1$ time slots where N is the number of channels.

Liu *et al* [18] employed the circular ring concept and devised ring-walk (RW) algorithm where potential channels are considered as vertices in the circular ring. SUs walk along the vertices with different velocities. SUs with lower velocities are expected to be caught by SUs with higher velocities. However, dependence of RW on the number of SUs may limit the network size.

Theis *et al* [19] developed the modular clock (MC) algorithm by leveraging the property of prime number modular arithmetic. The main driving factor of MC is the rate r at which SUs hop between the channels. MC guarantees the rendezvous only if SUs select different rates. Since rate is randomly chosen, MC failed to provide guaranteed rendezvous among SUs.

Liu *et al* [20] proposed the jump-stay (JS) algorithm, which guarantees the rendezvous among SUs both in symmetric and asymmetric models. JS consists of two phases: jump and stay. During the jump phase, SUs hop among available channels for $2P$ time slots where P is the smallest prime number greater than the total number of channels for communication. Each jump phase is followed by a stay phase of P time slots where SUs wait on a particular channel that is determined by the rate r. If two SUs select different rates, then rendezvous is expected in the jump phase; otherwise, rendezvous occurs during the stay phase. In symmetric model, MTTR of JS is $3P$ time slots. Lin *et al* [21] extended the JS algorithm and proposed enhanced jump-stay (EJS), which has significant performance improvement in asymmetric scenarios. However, EJS does not exhibit further improvements in symmetric model.

Chuang *et al* [13] presented the alternate hop-wait (AHW) rendezvous algorithm with the basic idea that if one SU hops among potential channels in a circular manner (referred to as hop mode) and the other waits on a particular channel (referred to as wait mode), then rendezvous is guaranteed by the time the first one completes a round. CH sequence of a SU is determined by the respective bits of its unique ID. Later, Chuang *et al* [14] proposed an enhanced alternate hop-wait (E-AHW) algorithm, where MAC address of SU is exploited as the unique ID. In symmetric model, E-AHW exhibits MTTR of $147P$ time slots where P is the smallest prime number greater than the number of potential channels.

3. CASR algorithm

3.1 *System model*

In this paper, we consider a CRN consisting of N $(N > 1)$ SUs that coexist with a set of PUs. Potential licensed spectrum is divided into M $(M > 1)$ non-overlapping channels $C = \{c_1, c_2, \ldots, c_M\}$ that are uniquely indexed in the range 1, 2, 3, …, M. Each SU possesses a unique ID and is assumed to be equipped with a CR. SUs are able to opportunistically access free, available channels in C if not occupied by the PUs. Free, available channels are acquired through spectrum sensing. It is assumed that the same set of available channels is associated with a pair of SUs that strive for rendezvous. As mentioned previously, this type of model is referred to as symmetric model. A time-slotted system is considered, where time is divided into slots of equal size. SUs hop among available channels with one channel per time slot, i.e., SUs do not switch channels within a time slot. The length of each time slot is set to $2t$ as in IEEE 802.22 [22], where t is the amount of time required to establish communication links between SUs. With this scenario, consider two SUs (i.e., SU_A and SU_B) with similar available channels that attempt to rendezvous by hopping on available channels with one channel per time slot. The problem is to construct the CH sequences so that rendezvous can be guaranteed on commonly available channels between the SUs in finite time.

3.2 *Fundamental concept*

Using MC algorithm, it is proved that if CH sequence of SU_i at time slot $t+1$ is formulated as $j_i^{t+1} = (j_i^t + r_i) \bmod (p_i)$ and $r_i \neq r_j$ (i.e., different rates are selected by SUs) then rendezvous can be guaranteed within p_i time slots. However, if SUs select the same rate, then rendezvous cannot be guaranteed. Thus, for rendezvous guarantee, the rates selected by SUs should always be different. MC failed to achieve rendezvous guarantee as the rate is randomly chosen. The proposed approach ensures that SUs select different rates within a certain period of time. The gist of the proposed approach is the methodical utilization of unique IDs assigned to the SUs. CH sequences of SUs are fabricated in conformity with the bits of unique IDs. Usage of universal MAC address as unique ID ensures that the proposed approach can discern nearly every network device. Following the proposed approach, SUs succeed in selecting different rates within a certain bounded time period by employing their unique IDs. The bits of unique IDs are used to generate the rates at which SUs hop among channels. The ID bits are rotated accordingly when all bits are employed.

3.3 *Algorithm description*

Algorithm 1 Comprehensive asynchronous symmetric rendezvous algorithm

```
Input: number of available channels, unique ID.
Output: channel hopping sequence.
calculate $p$, the next prime $\geq m$
$j^0 \leftarrow$ initial channel index, choose randomly in $[0, m)$
calculate $len$, the length of $uid$
$ptr \leftarrow 1, itr \leftarrow 1$
$IndexTable \leftarrow distribute\_BitsIntoGroups(len, p)$
while not rendezvous do
    if $itr \leq (len/2)$ then
        $r \leftarrow selectRate(IndexTable, ptr, uid)$
        for $t = 0$ to $2p - 1$ do
            $j^{t+1} \leftarrow (j^t + r) \mod (p)$
            if $j^{t+1} < m$ then
                $c \leftarrow c_{j^{t+1}}$
            else
                $c \leftarrow c_{j^{t+1} \mod m}$
            end if
            attempt rendezvous on channel $c$
        end for
        increment $ptr$ and $itr$ by 1.
    else
        for $t = 0$ to $2p - 1$ do
            $r \leftarrow selectRate(IndexTable, ptr - 1, uid)$
            $c \leftarrow r \mod m$
            attempt rendezvous on channel $c$
        end for
        $itr \leftarrow 1$
    end if
    if $ptr > len$ then
        $ptr \leftarrow 1$
    end if
end while
```

Algorithm 2 *distribute_BitsIntoGroups(len, p)*

```
$len$ is the length of unique ID bit sequence
$p$ is the prime number
$g$ is the number of groups
$x$ is the number of bits per group
$y$ is the number of groups with additional bit
$IndexTable(ptr, index)$ is a table for each bit and its corresponding group index where $ptr$ is the index to the bit position in the unique ID and $index$ is the group index, initially $ptr = 1$, $index = 0$.
$g = (p - 1)/2$
if $len > g$ then
    $x = len/g$
    $y = len \mod g$
else
    $x = 1$
    $y = 0$
end if
$k = x \times (g - y)$
$limit = x$
while $ptr \leq len$ do
    $IndexTable(ptr) = index$
    if $ptr \geq limit$ then
        $index = index + 1$
        $limit = limit + x$
        if $ptr \geq k$ then
            $limit = limit + x$
        end if
    end if
    $ptr = ptr + 1$
end while
```

Algorithm 3 *SelectRate(IndexTable, ptr, uid)*

```
$index$ is the group number in which the bit at $ptr$ belongs to, $bitvalue$ is the value of the bit at the position $ptr$ in $uid$, $r$ is the rate used for channel switching.
$index = IndexTable(ptr)$
$bitvalue$ = the value of the bit at position $ptr$ from $uid$
$r = bitvalue + (2 \times index) + 1$
```

CASR algorithm (outlined in Algorithm 1) is executed whenever a node (SU) starts the rendezvous process. Important parameters steering the algorithm are as follows:

- *uid* refers to the unique ID bit sequence,
- *len* is the length of *uid*,
- r is the rate at which SUs hop among channels,
- t refers to the time slot of the system,
- m is the number of available channels that are acquired through spectrum sensing and
- p is the smallest prime number greater than or equal to m.

CASR algorithm begins by calculating $p \geq$ number of available channels m, initial channel index j^0 randomly from the set of available channels in [0, m), and the length of uid in bits. The algorithm then divides the bits of uid into various logical groups and constructs an index table that maps each bit of the uid to its corresponding group index. The index table is used by the function *selectRate*() to determine the rate r to be used for switching among channels. Next, the algorithm enters to the hop/jump state for $n = len/2$ iterations during which SU hops on available channels at the same rate for an iteration of $2p$ time slots. The channel index j is increased by r mod p after each time slot t. If the channel index j is in [0, m), then the radio moves to channel c_j, i.e., the channel with index j in the set of available channels. Otherwise, remapping is done via the mod operation to obtain the channel in [0, m) and the radio switches to that channel. The algorithm maintains the same rate r for an iteration of $2p$ time slots. This ensures an overlap of at least p time slots between the SUs without changing the rates. Consecutive p time slots cover all m

channels. Thus, for $2p$ time slots, SU switches among all the available channels with the same r value. After every n iterations, the algorithm introduces a stay period of $2p$ time slots, where SU waits on the channel that is determined by the previous r value. When MAC address is used as the unique ID, $n = 48/2$ and hence the stay period is incorporated after every 24 iterations. After the stay period, the algorithm re-enters the hop-jump state, thus continuing the cycle. The algorithm is terminated once rendezvous is achieved between the SUs.

The CASR algorithm presents a novel grouping concept where bits of unique ID are logically categorized into various groups. Logical grouping ensures the selection of different rates by the SUs within the bounded time. Grouping is carried out according to the number of channels m and the prime number p. Number of groups, g is calculated as

$$g = (p - 1)/2 \tag{1}$$

where p is the smallest prime number $\geq m$. If the system employs some local unique identifier instead of universal MAC address then the length of *uid* may be less than the number of groups g. In such cases, only one bit exists in each group. Otherwise, the number of bits per group is calculated as

$$x = len/g. \tag{2}$$

If possible, bits are evenly distributed among the groups. Otherwise, extra bits are incorporated in the groups starting from the most significant bit (MSB) towards the least significant bit (LSB) by adding one extra bit to those groups. The number of groups having such extra bit, y is calculated as $y = len \bmod g$. Table 1 shows the number of groups for different number of available channels. The maximum number of bits in the group is referred to as *glen* (group length). Then for each bit in the *uid*, the algorithm finds out its group index and creates a Table *IndexTable (ptr, index)* that maps each bit of the *uid* at the position *ptr* to its corresponding group index.

Before starting each iteration of $2p$ time slots, the algorithm invokes the function *selectRate*(*IndexTable*, *ptr*, *uid*) to determine the rate to be used for channel switching. The function calculates the rate r as

$$r = bitvalue + (2 \times index) + 1 \tag{3}$$

where *bitvalue* is the value of the bit at position *ptr* in *uid*, and *index* is the group number in which the bit at *ptr* belongs to. The *bitvalue* can be either 0 or 1. Since the rate r is calculated as a function of *bitvalue* as well as *index*, it is ensured that the rates generated by the bits from two different groups will never be the same. An example of logical grouping when $m = 10$ is illustrated in figure 2. The number of groups g in this case is 5 and group index varies from 0 to 4. Different groups are shown using bidirectional arrows. Since each bit can have two values (i.e., either 0 or 1), there will be two possible rates for each group. The possible rates generated by different groups are mentioned on the top of each group. Hence, as mentioned, the algorithm ensures the existence of at least two groups and there will not be any overlap in the rate values generated by two different groups.

Table 1. Number of available channels and grouping details.

m	p	No. of groups g	No. of bits/group
5	5*	2	24
10	11	5	9 or 10
20	23	11	4 or 5
50	53	26	1 or 2
75	79	39	1 or 2
90	97	48	1
100	101	50	1

Remarks: *If $m < 4$, we consider $p = 5$ as a special case (as per definition, it should be smallest prime $\geq m$) for ensuring the existence of more than one group

4. Performance evaluation

The main determinant of CASR algorithm is the rate r (chosen by the SU) which directly relies on the bits of the unique ID and the prime number p. CASR algorithm ensures the rendezvous between two SUs within p time slots under the constraint that the rates selected by the two SUs should be different. When the rates are taken according to the bits of the MAC address, we can ensure the selection of different rates within 48 iterations provided there exists some special interpretation for MSB or LSB to avoid the rotation problem. In order to optimize this interval, CASR algorithm provides different interpretations for bits to form different groups. Hence, the likelihood of choosing different rates can be broadly categorized as follows:

1. The corresponding ID bits of SUs are different (i.e., one SU uses a bit of value 1 and the other SU uses 0 or vice versa).
2. SUs access different groups simultaneously.

Theorem 1 *In asynchronous scenario, when there exist a delay of at least one bit among two SUs A and B, then the maximum time required for rendezvous is $2p \times glen$.*

Proof As each SU possesses a unique identifier, there will be at least one bit difference in the IDs of any two SUs. Different possibilities in the rate selection by the SUs are illustrated in figure 3. The shaded portion represents the iteration where both SUs select different r and hence

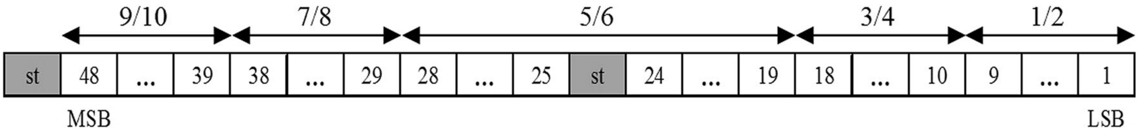


Figure 2. Logical grouping of unique ID.

rendezvous. Empty portions represent the lag between the SUs in terms of bit position. Rate selection by two SUs when the IDs differ in the first bit (LSB) is shown in figure 3.1. The worst case scenario is chosen where the IDs differ by exactly one bit. The groups made out of the unique ID vary according to the number of channels. Three cases are shown for different values of available channels. A lag of 1 bit position is shown in figure 3.1.b under different cases. Figure 3.2 describes the scenarios when the IDs differ at the last bit.

Case 1: $m = 10$, when one SU reads the 10th bit, the other SU would be in the 9th bit. Here 9th bit belongs to group 0 and 10th bit belongs to group 1 and hence generate different rates. Thus, rendezvous can be ensured in 9th iteration of the second SU.

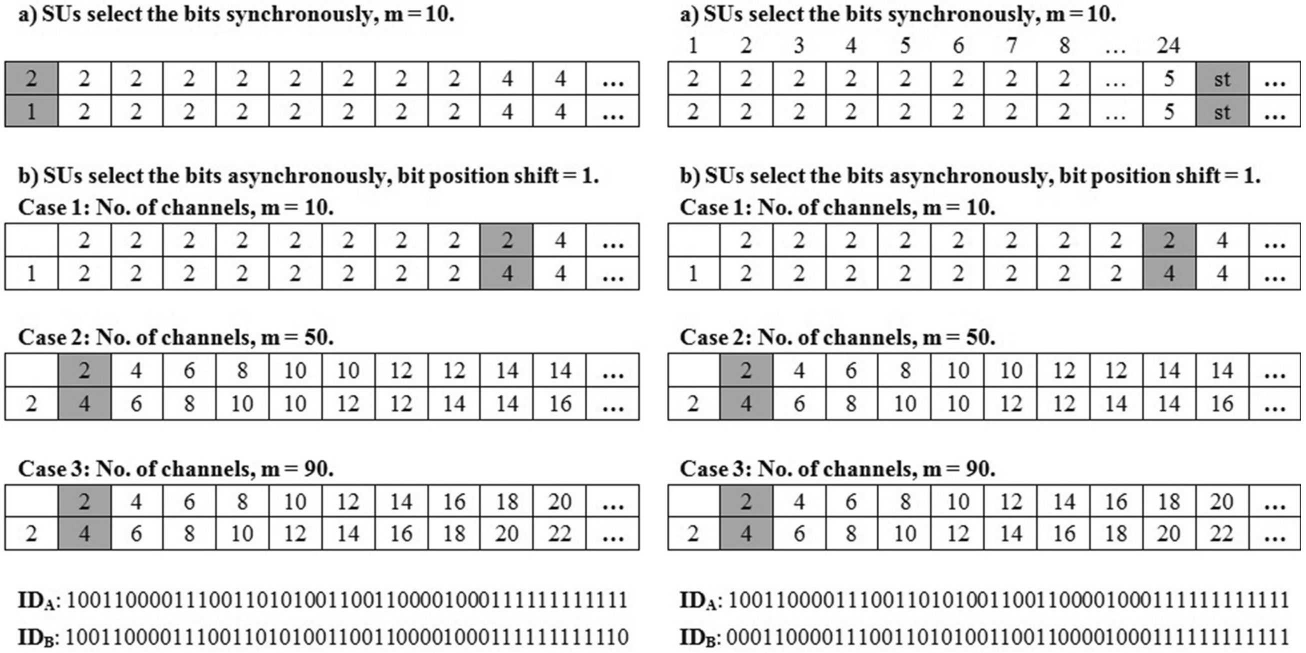
Case 2: $m = 50$, when one SU reads the 2nd bit, the other SU would be in the 1st bit. Since the first four groups contain only one bit each, 1st and 2nd bits belong to different groups and hence result in generating different rates. Therefore rendezvous is confirmed in the first iteration.

Case 3: $m = 90$, when the second SU reads the 1st bit, first SU would be in the 2nd bit. Since all the groups contain only one bit each, 1st and 2nd bits belong to different groups and hence result in generating different rates. Here rendezvous is ensured in the first iteration.

Similarly, if there is a shift of at least one bit among the SUs, selection of different rates by two SUs can be ensured within certain bounded iterations determined by the group length regardless of the difference in ID bits. □

Theorem 2 *The upper bound of ETTR is 2.05 p and hence, on average, rendezvous between the two SUs A and B nearly happens within the first iteration itself.*

Proof Rendezvous is expected to occur when the SUs choose different rates simultaneously, having an overlap of at least p time slots. As per the algorithm, for instance, there exist two groups if $m = 5$. The bit employed by the SUs to generate the rate can be from any of the two groups. When the SUs select the bits from two groups, four possible combinations arise. If the bit used by the SUs falls into different groups, then the generated rates



a) SUs select the bits synchronously, m = 10.

2	2	2	2	2	2	2	2	2	4	4	…
1	2	2	2	2	2	2	2	2	4	4	…

b) SUs select the bits asynchronously, bit position shift = 1.

Case 1: No. of channels, m = 10.

	2	2	2	2	2	2	2	2	2	4	…
1	2	2	2	2	2	2	2	2	4	4	…

Case 2: No. of channels, m = 50.

	2	4	6	8	10	10	12	12	14	14	…
2	4	6	8	10	10	12	12	14	14	16	…

Case 3: No. of channels, m = 90.

	2	4	6	8	10	12	14	16	18	20	…
2	4	6	8	10	12	14	16	18	20	22	…

ID$_A$: 100110000111001101010011001100001000111111111111

ID$_B$: 100110000111001101010011001100001000111111111110

Figure 3.1: Unique IDs of A and B differ in the first bit.

a) SUs select the bits synchronously, m = 10.

1	2	3	4	5	6	7	8	…	24		
2	2	2	2	2	2	2	2	…	5	st	…
2	2	2	2	2	2	2	2	…	5	st	…

b) SUs select the bits asynchronously, bit position shift = 1.

Case 1: No. of channels, m = 10.

	2	2	2	2	2	2	2	2	2	4	…
1	2	2	2	2	2	2	2	2	4	4	…

Case 2: No. of channels, m = 50.

	2	4	6	8	10	10	12	12	14	14	…
2	4	6	8	10	10	12	12	14	14	16	…

Case 3: No. of channels, m = 90.

	2	4	6	8	10	12	14	16	18	20	…
2	4	6	8	10	12	14	16	18	20	22	…

ID$_A$: 100110000111001101010011001100001000111111111111

ID$_B$: 000110000111001101010011001100001000111111111111

Figure 3.2: Unique IDs of A and B differ in the last (48th) bit.

Figure 3. Different possibilities in the rate selection by two SUs.

will be different. Hence with 1/2 probability, both will select the same group. Further, there are two possible rates in every group depending on whether the bit is 0 or 1, generating a combination of 2^2. Therefore, the probability of choosing the same rate within a group is $\frac{2}{4} = \frac{1}{2}$ and the probability of selecting the same rate is $k = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$. Hence, the SUs select different rates with $1 - k$ (i.e., $\frac{3}{4}$) probability and rendezvous within $2p$ time slots. Similarly, with probability $\frac{1}{4} \times \frac{3}{4}$, the SUs will rendezvous within $4p$ time slots. After 24 iterations, rendezvous is guaranteed and the probability that SUs do not rendezvous will turn to zero. As the number of channels m increases, the number of groups increases, which further increases the probability of choosing different rates. Hence, the ETTR can be formulated as follows:

$$ETTR \leq \frac{1}{M} \times \sum_{m=1}^{M} \sum_{x=1}^{24} 2px(1-k)(k)^{x-1} \tag{4}$$

where p is the smallest prime number $\geq m$, $g = (p-1)/2$, M is the limit of the maximum number of possible channels, $k = g/(2 \times 2^g)$. On evaluating this expression up to the limit $M = 100$, ETTR is calculated to be $\leq 2.05p$. As the number of available channels increases, so does the number of groups and the probability of choosing different rates by the SUs. Hence, further increase in the maximum number of possible channels will result in a better ETTR. □

Theorem 3 *If two SUs A and B synchronously select the bits and the difference in their IDs appears after* 24*th bit then rendezvous is ensured within the duration of* 50p *time slots.*

Proof When the two SUs choose the bits synchronously and the difference in the IDs comes only after the 24 bits (illustrated in figure 3.2a), selection of different rates cannot be ensured within the 24 iterations. As stated in Section 3.2, after every n iteration, a stay period is introduced, where $n =$ (length of ID bit sequence)/2. Hence, when MAC address is used, $n = 48/2$ and the stay period is incorporated after every 24 iterations. In this case, SUs synchronously select the bits and there is no difference in IDs till the 24th bit. This indicates that the values of r selected by the SUs are the same till the 24th iteration. Therefore, when SUs enter the 25th iteration, the previous r values of the SUs are the same. Now, during the stay period, both SUs wait on a channel determined by the previous r values, for the duration of $2p$ time slots. Since the previous r values of the SUs are the same, both SUs stay on the same channel for the duration of $2p$ time slots. This ensures rendezvous between the SUs in the 25th iteration as the SUs stay on the same channel for the duration of $2p$ time slots. Since each iteration constitutes $2p$ time slots, MTTR turns out to be $25 \times 2p = 50p$ time slots. □

5. Simulation and comparative analysis

Rendezvous process simulation requires an integrated programming environment capable of handling numerical computing and user interfacing features. The tools that are used to investigate the performance of CASR algorithm include Matlab and C programming. We implemented CASR algorithm in Matlab 7.11 and performed several simulation experiments. Simulation scenarios are built by defining the CRN environment in terms of the existence of PUs and SUs and their network attributes. This includes defining the basic parameters such as the number of PUs and SUs, transmission range of the nodes and the observed available channels of SUs. During the simulations, pairwise rendezvous between SUs has been considered in the symmetric model. Thus, observed channels of two SUs during the rendezvous process are the same. The geometrical locations of SUs were randomly generated and the number of PUs and SUs are defined by the input parameters. Primary attributes of the rendezvous process include the number of available channels m and the unique ID of SUs. Another factor determining the outcome is the time difference t between the SUs. Due to the asynchronous nature, SUs may employ different bits (i.e., bits in different positions) or bits may be in different positions within the iteration of $2p$ time slots. All these factors have been considered during the simulations and TTR is measured in each execution of the rendezvous process. ETTR is used as an important metric to determine the stability and consistency of CASR algorithm. Each simulation experiment is executed and averaged over 1000 independent runs in order to derive the most appropriate value of ETTR.

Simulation experiments are conducted primarily in two different categories. In the first case, rendezvous parameters are generated on purely random basis. In the second case, simulations are executed by explicitly including the most unfavourable cases. When algorithm parameters are generated randomly, the performance of CASR algorithm is much better than the theoretical results. Figure 4 shows the average TTR for different values of m. As m goes higher, the radio has to scan more channels and hence the TTR also rises. Further, to ensure the impact of worst-case scenarios, simulations are executed by explicitly including the most unfavourable cases. Figure 5 illustrates the result of the simulations. By combining the result of these two scenarios (shown in figures 4 and 5), average TTR is calculated to be $0.95p$. Hence, the result affirms the theoretical conclusion and shows considerable improvement over the theoretically estimated value.

Further, the behaviour of CASR algorithm is tested against varying length of the unique ID. To study the impact of ID length on TTR, CASR algorithm is simulated with unique IDs of different lengths. The variation of ETTR under the different lengths of unique ID is shown in figure 6. The value of m is taken as 50. From figure 6, it can

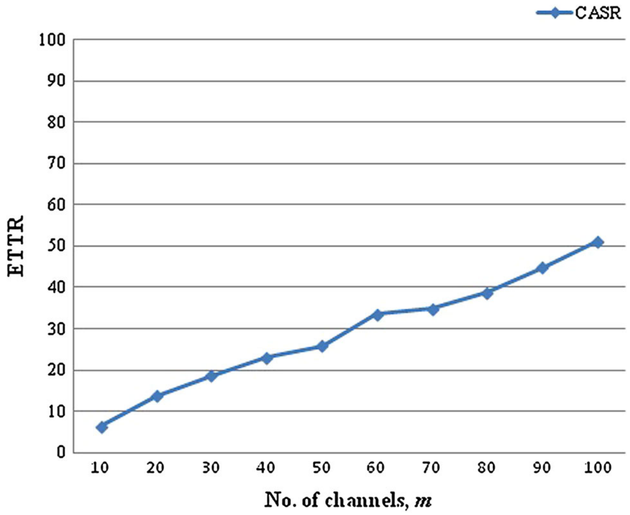


Figure 4. Variation of average TTR in random scenario.

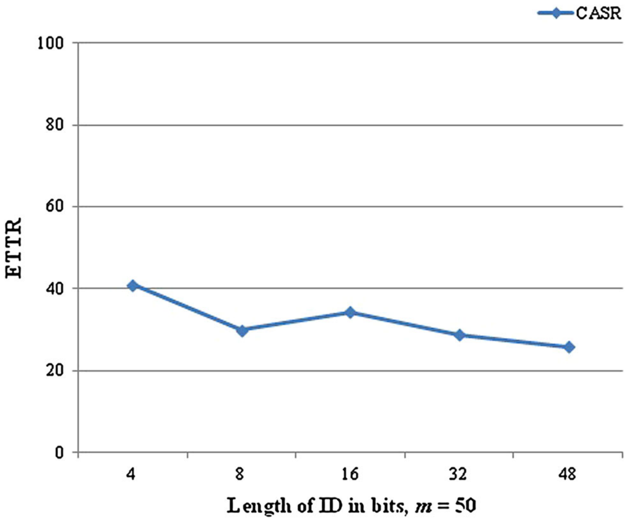


Figure 6. Variation of average TTR against different ID lengths (in bits).

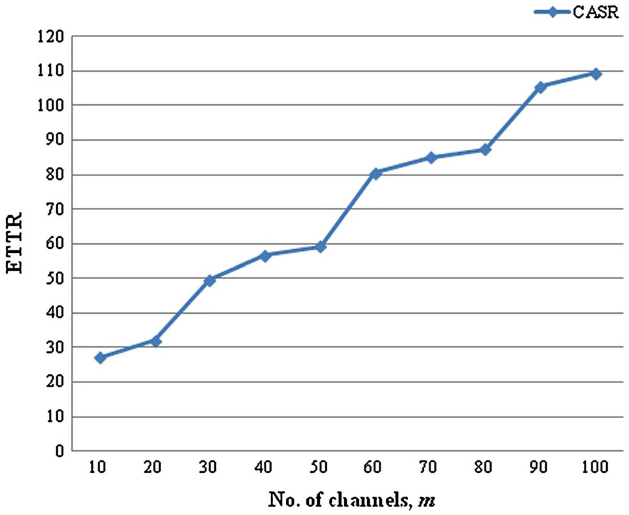


Figure 5. Variation of average TTR in explicit scenario.

be observed that ETTR does not have a drastic variation on increasing the length of the unique ID. CASR algorithm shows a higher ETTR when the length of ID is 4. This is because of the fact that as the length of the ID increases, the increase in MTTR is balanced by the corresponding reduction in ETTR.

In order to further evaluate the performance of CASR against existing rendezvous methods, we implemented CASR algorithm and some state-of-the-art rendezvous algorithms (i.e., AHW [13], E-AHW [14], RW [18], JS [20], EJS [21], S-ACH [23] and DRDS [24]) using C programming and performed several simulations under different environments. Simulation results of CASR are then compared to the performance of other rendezvous algorithms under a similar environment. In the simulations, symmetric model has been considered, where the SUs have similar available channels. The total number of SUs is taken to be 100. The length of all ID-based algorithms is set to 48. Each algorithm is simulated for 1000 independent runs with varying number of available channels and the final ETTR is calculated by averaging the ETTR of 1000 runs. The results of comparisons are illustrated in figure 7.

First, the performance is studied with respect to the increase in the number of SUs (i.e., network size). The same scenario is executed with 10 and 100 SUs. It can be observed from figure 7a that the network size slightly increases the ETTR of S-ACH and has little or no effect on the ETTR of E-AHW and CASR. This is due to utilization of universal MAC address as unique ID for generating the CH sequence. However, RW exploits the unique identity of SU, which increases with the number of SUs. Hence, ETTR of RW is significantly influenced by the network size. Another simulation is carried out with the increase in the number of channels in the network. It can be observed from figure 7b that the number of channels has a significant impact on the ETTR of all algorithms. S-ACH and RW are heavily affected by the number of channels as both algorithms are highly dependent on the number of channels. ETTRs of DRDS and CASR are nearly similar for lesser number of channels. However, the performance of CASR is better with the increase in the number of channels. This is due to comparatively lesser ETTR of CASR as compared with DRDS. Thus, if the number of channels is more, the performance of CASR is significantly better than those of the other algorithms.

The performance of CASR is also compared to those of JS and EJS, which are known to be the most prominent rendezvous algorithms proposed for CRNs. The simulation results are illustrated in figure 7c. As shown in the figure, ETTRs of JS, EJS and CASR are almost the same with

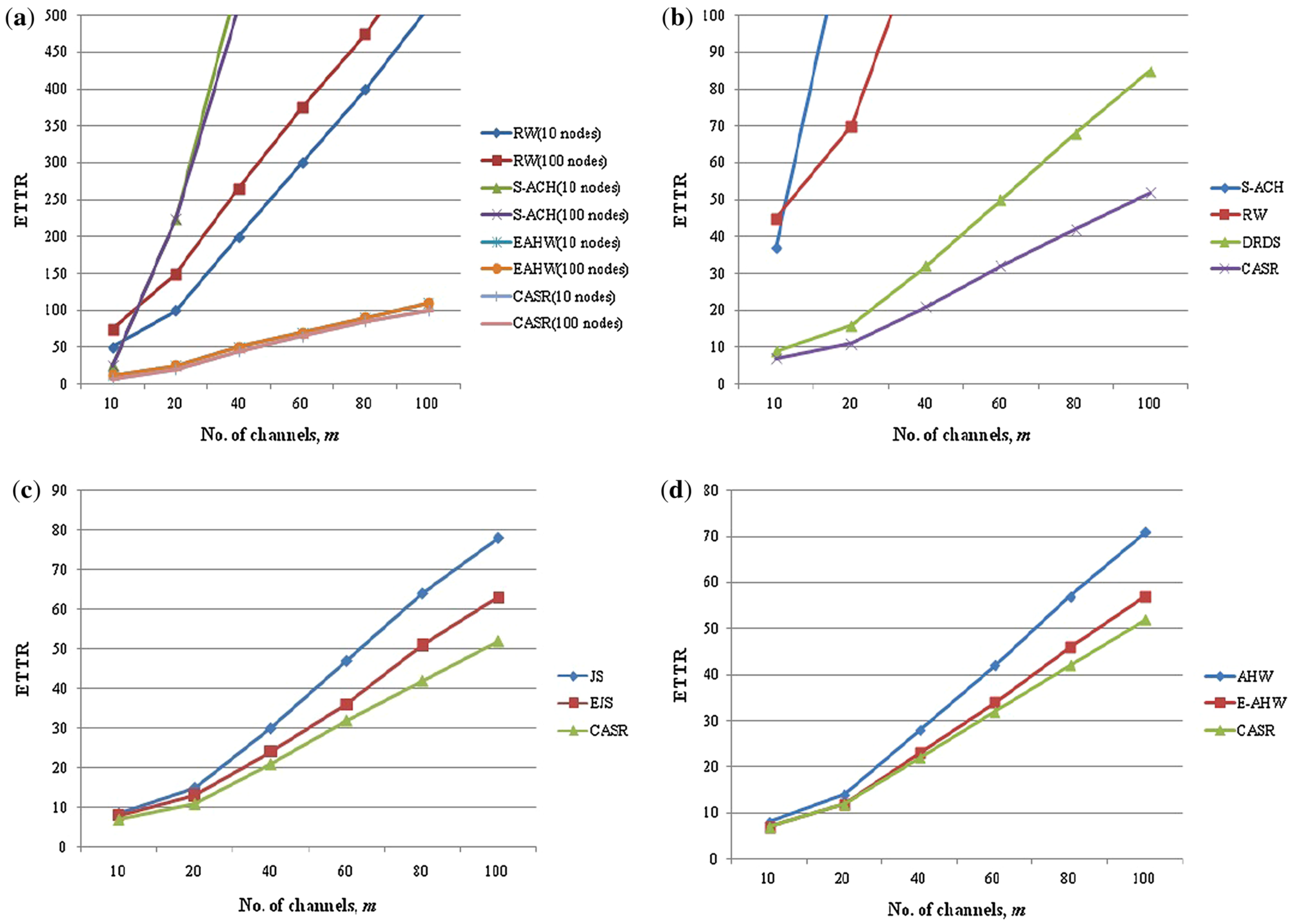


Figure 7. Average TTR comparison in symmetric model.

lesser number of channels. However, CASR performs better than JS and EJS with higher number of channels as the increase in number of channels increases the probability of choosing different rates by the SUs, which further increases the rendezvous probability. JS is significantly affected by the increase in the number of channels. Finally, ETTR of CASR is compared to ETTR of AHW and E-AHW and the result is plotted in figure 7d. From the figure, it is evident that AHW, E-AHW and CASR have nearly similar ETTR with lower number of channels. However, CASR exhibits comparatively lesser ETTR especially when the number of channels is higher. This is also due to the increase in rendezvous probability with increasing number of channels. For instance, when the number of channels is 20 or below, AHW, E-AHW and CASR algorithms show nearly the same ETTR. However, when the number of channels is 50 or above, CASR exhibits a lower ETTR as compared with AHW and E-AHW. Furthermore, MTTR of E-AHW on using the MAC address is $147P$ and ETTR is $\sim 13P/6$, which is greater than the estimated MTTR and ETTR of CASR.

6. Conclusion and future work

The ultimate objective of the paper was to achieve pairwise rendezvous between SUs in CRN under the symmetric model. An ID-based rendezvous algorithm named CASR is proposed that guarantees rendezvous of SUs in finite time without the requirement of time synchronization. CASR algorithm exploits the MAC address as unique ID to contrive the CH sequence. Unique ID is dynamically manipulated in accordance with the list of free, available channels. Construction of CH sequence is coupled with the ID manipulation. The primary focus of CASR algorithm is to reduce the average TTR (ETTR). Efficiency of CASR algorithm is theoretically estimated and verified through practical simulation experiments. Simulation results affirm that CASR algorithm is more viable and results in better ETTR as compared with the existing rendezvous algorithms. Potential future work includes the extension of CASR that befits for symmetric as well as asymmetric models.

References

[1] Haykin S 2005 Cognitive radio: brain-empowered wireless communications. *IEEE J. Select. Areas Commun.* 23: 201–220

[2] Cabric D, Mishra S M, Willkomm D, Brodersen R W and Wolisz A 2005 A cognitive radio approach for usage of virtual unlicensed spectrum. In: *Proceedings of the 14th IST Mobile and Wireless Communications Summit*, pp. 1–4

[3] Thomas R, DaSilva L and MacKenzie A 2005 Cognitive networks. In: *Proceedings of the First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN*

[4] Akyildiz I, Lee W, Vuran M and Mohanty S 2006 NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey. *Comput. Netw.* 50: 2127–2159

[5] Jondral F K 2005 Software-defined radio-basics and evolution to cognitive radio. *EURASIP J. Wirel. Commun. Netw.* 3: 275–283

[6] Akyildiz I, Lee W and Chowdhury K 2009 CRAHNs: cognitive radio ad hoc networks. *Ad Hoc Netw.* 7: 810–836

[7] Sengupta S and Subbalakshmi K 2013 Open research issues in multi-hop cognitive radio networks. *IEEE Commun. Mag.* 51: 168–176

[8] Liu H, Lin Z, Chu X and Leung Y 2012 Taxonomy and challenges of rendezvous algorithms in cognitive radio networks. In: *Proceedings of the International Conference on Computing, Networking and Communications (ICNC)*

[9] Shin J, Yang D and Kim C 2010 A channel rendezvous scheme for cognitive radio networks. *IEEE Commun. Lett.* 14: 954–956

[10] Lo B 2011 A survey of common control channel design in cognitive radio networks. *Phys. Commun.* 4: 26–39

[11] Joshi G P, Nam S Y and Kim S W 2014 Rendezvous issues in ad hoc cognitive radio networks. *KSII Trans. Internet Inform. Syst. (TIIS)* 8: 3655–3673

[12] Chang G, Teng W, Chen H and Sheu J 2014 Novel channel-hopping schemes for cognitive radio networks. *IEEE Trans. Mobile Comput.* 13: 407–421

[13] Chuang I, Wu H, Lee K and Kuo Y 2013 Alternate hop-and-wait channel rendezvous method for cognitive radio networks. In: *Proceedings of IEEE INFOCOM*

[14] Chuang I, Wu H and Kuo Y 2014 A fast blind rendezvous method by alternate hop-and-wait channel hopping in cognitive radio networks. *IEEE Trans. Mobile Comput.* 13: 2171–2184

[15] Cormio C and Chowdhury K 2010 Common control channel design for cognitive radio wireless ad hoc networks using adaptive frequency hopping. *Ad Hoc Netw.* 8: 430–438

[16] Kondareddy Y, Agrawal P and Sivalingam K 2008 Cognitive radio network setup without a common control channel. In: *Proceedings of the 2008 IEEE Military Communications Conference, MILCOM*

[17] Yang D, Shin J and Kim C 2010 Deterministic rendezvous scheme in multichannel access networks. *Electron. Lett.* 46: 1402

[18] Liu H, Lin Z, Chu X and Leung Y 2010 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 & International Conference on Cyber, Physical and Social Computing*

[19] Theis N, Thomas R and DaSilva L 2011 Rendezvous for cognitive radios. *IEEE Trans. Mobile Comput.* 10: 216–227

[20] Liu H, Lin Z, Chu X and Leung Y 2012 Jump-stay rendezvous algorithm for cognitive radio networks. *IEEE Trans. Parallel Distrib. Syst.* 23: 1867–1881

[21] Lin Z, Liu H, Chu X and Leung Y 2013 Enhanced jump-stay rendezvous algorithm for cognitive radio networks. *IEEE Commun. Lett.* 17: 1742–1745

[22] Stevenson C R, Chouinard G, Lei Z, Hu W, Shellhammer S J and Caldwell W 2009 IEEE 802.22: the first cognitive radio wireless regional area network standard. *IEEE J. Mag.* 1: 130–138

[23] Bian K and Park J-M J 2013 Maximizing rendezvous diversity in rendezvous protocols for decentralized cognitive radio networks. *IEEE Trans. Mobile Comput.* 12: 1294–1307

[24] Gu Z, Hua Q, Wang Y and Lau F 2013 Nearly optimal asynchronous blind rendezvous algorithm for cognitive radio networks. In: *Proceedings of the IEEE International Conference on Sensing, Communications and Networking (SECON)*