Dynamic Spectrum Allocation Algorithms for Industrial Cognitive Radio Networks

Prasan Kumar Sahoo, *Senior Member, IEEE*, Sulagna Mohapatra, and Jang-Ping Sheu, *Fellow, IEEE*

***Abstract*—Irregular spectrum usage and spectrum scarcity in emergency situations is a common problem in industrial wireless networks. To enhance the dynamic spectrum usage, cognitive radio network (CRN) is introduced in various automotive industrial wireless applications named as industrial cognitive radio network (ICRN). However, establishing the control channel by using channel hopping mechanism in ICRN is a challenging problem. In order to achieve reliable performance in ICRN, efficient channel hopping protocols need to be designed. In this paper, two channel hopping protocols are designed for the ICRN with or without the global clock synchronization to maximize the degree of rendezvous within the shortest time, minimize the inter rendezvous intervals and to reduce the maximum time to rendezvous by two secondary users. Performance evaluation of our protocols outperform in terms of throughput, percentage of rendezvous and average time to rendezvous over existing CRN protocols.**

***Index Terms*—Asynchronous, channel hopping (CH), cognitive radio networks (CRNs), symmetric, synchronous.**

I. Introduction

COGNITIVE radio network (CRN) is introduced by Federal Communications Commission [1] to acknowledge the spectrum scarcity problems and to enhance the dynamic spectrum access (DSA) [2]. The main objective of CRN is to introduce some unlicensed users known as secondary users (SUs) to utilize the unused spectrum in the absence of the licensed users known as primary users (PUs) [3], [4]. Each SU is equipped with one or more cognitive radios [5], [6] and is capable of identifying the availability of channels not occupied by the PUs. The entities of a CRN are the SUs, which can adequately sense the spectrum holes to rendezvous with other SUs and establish the common control channel for transmitting data. Each SU executes the channel hopping (CH) approach for the purpose of rendezvous to use the underutilized spectrum efficiently.

Wireless technologies, such as ZigBee, WiFi, Bluetooth, and 5G/LTE play an important role in various industrial applications in process controlling, monitoring, surveillance, and other connected embedded systems. In such technological environment of industrial wireless networks, multiple heterogeneous wireless devices may operate in a frequency band utilized by many other wireless technologies, which are usually harsh and time varying. Based on the current regulation of radio spectrum for the industrial wireless devices, the spectrum assignment is fixed for which devices cannot switch to other spectrum even if it is congested [7] at particular regions. Besides, some part of the allocated spectrum in industry are overutilized, whereas some part of it are underutilized, which creates spectrum holes based on time and space. In order to mitigate such problems, cognitive radio technology should be integrated to the industrial wireless devices for spectrum sensing and DSA, which is called industrial cognitive radio network (ICRN).

The fire alarm notification in industries requires quick delivery of messages. If the associated licensed spectrum is not available at that time, the entire factory floor might be under fire. Due to scarcity of the communication channels, if the data are received at wrong instance of time, the situation might lead to inaccurate decision of monitoring system [3]. Considering the harsh and emergency environment of the industries, ICRN needs to be embedded with much hardened and smarter cognitive antennas and devices as compared to the devices of general CRN. ICRN can use the cognitive radios to sense the spectrum holes in the emergency situations to transmit data. In an industry, normally WiFi and cellular network are busy and are overutilized, whereas the spectrum assigned for the satellite communication is underutilized. In this scenario, concept of CRN can be used in industries by introducing the unlicensed/secondary WiFi or cellular users to use the vacant portion of the satellite spectrum for emergency data communication.

In wireless networks, though CH multichannel protocols can be used to establish the common control channel for the medium access, the mechanism is quite different from the CRN in terms of operational and sensing capability. Unlike CRN, the wireless communication devices do not have cognitive radios to sense the unused spectrum. Moreover, the devices in wireless net-

Manuscript received September 15, 2017; accepted October 27, 2017. Date of publication November 16, 2017; date of current version July 2, 2018. This work was supported by the Ministry of Science and Technology, Taiwan, under Grant 106-2221-E-182-014 and 106-2221-E-007-019-MY3. Paper no. TII-17-2152. *(Corresponding author: Sulagna Mohapatra.)*

P. K. Sahoo is with the Department of Computer Science and Information Engineering, Chang Gung University, Guishan 33302, Taiwan, and also with the Division of Cardiology, Department of Internal Medicine, Chang Gung Memorial Hospital, Linkou 33305, Taiwan (e-mail: pksahoo@mail.cgu.edu.tw).

S. Mohapatra is with the Department of Computer Science and Information Engineering, Chang Gung University, Guishan 33302, Taiwan (e-mail: d0521007@stmail.cgu.edu.tw).

J.-P. Sheu is with the Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan (e-mail: sheujp@cs.nthu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2017.2774240

work operate on Industrial, Scientific and Medical (ISM) bands and there is no distinction between the devices as primary or secondary as all devices have equal rights over the channels to access. However, in CRN the incumbent users operated over the licensed spectrum are PUs and always have the first priority to use the channels. Each SU executes the CH approach in absence of the PUs, where it needs to hop from one channel to another to find the common vacant channel for making their rendezvous successful [8], [9].

SUs have to preempt the channels as soon as a PU returns to use it and the SU has to hop to another channel to find out the unused channel at different time instance [10]. The primary challenge in the CH approach is how to increase the degree of rendezvous, including minimization of the maximum time to rendezvous (MTTR) between SUs [11]–[13]. In this paper, symmetric synchronous *(*SymSyn) CH and symmetric asynchronous *(*SymAsyn) CH protocols are proposed to maximize the degree of rendezvous as well as to minimize MTTR with maximum channel utilization. It is to be noted that throughout this paper, CRN is alternatively used to represent ICRN.

Rest of the paper is organized as follows. Related works with our motivation and contributions are given in Section II. System model of the proposed CH protocol is given in Section III. The proposed symmetric synchronous and asynchronous CH protocols are designed in Section IV. Performance evaluation of our protocols is given in Section V and concluding remarks are made in Section VI.

II. Related work

In CH approach, network can be considered as *synchronous or asynchronous* and the channel type can be *symmetric or asymmetric*based on the availability of channels for the SUs coexist with the PUs [11]–[13]. According to the definition of the *synchronous* environment, all SUs can enter into the network at the same instance of time, whereas in *asynchronous* scenario SUs can enter into the network at any point of time [14]–[16]. In symmetric CH approach, each SU can have same set of available channels located in the same geographical area, whereas in asymmetric CH approach, each SU can have different sets of available channels with at least one common channel among them [13], [17]. Wu *et al.* [12], propose PRICH (round-robin indemnity-CH) and CACH (cycle-adjustable CH) in synchronous and symmetric environment. However, the degree of rendezvous in both protocols is very small, which is N out of $N \times (N+1)$ time slots, where N is the number of available channels for SUs.

Chang *et al.* [1] propose rendezvous couple CH (RCCH) scheme, which can increase the channel utilization ratio, but the degree of rendezvous is very small, i.e., N and the value of MTTR is large, i.e., $\frac{N}{2}$. In [1], another CH scheme is introduced as asynchronous rendezvous CH (ARCH) where probability of rendezvous is very less, i.e., $\frac{1}{N}$. Beside, MTTR is very large, which is $(2N-1)$. In [11], Asyn-ETCH protocol is designed for the asynchronous environment. However, it can have small degree of rendezvous, i.e., $\lessapprox 50\%$ with large MTTR $\lessapprox 2N$.

In [14], A-QCH (asynchronous quorum based CH system) algorithm is proposed where the number of rendezvous channel is always two. Beside, in the CH system A-MOCH [14], the degree of rendezvous is only N out of N^2 time slots. The rendezvous is not uniform in each cycle and some of the cycles are over- or underutilized. In [15], enhanced alternate hop and wait (E-AHW) CH algorithm is proposed for asynchronous symmetric environment. In E-AHW, average degree of rendezvous is only 15%, which occurs only in a single channel.

According to [13], the MTTR between two SUs in SymAsyn scenario is $3P$, where P is a prime number and $P > N$. In other words, two SUs need to wait for an entire inner round to have the rendezvous. In [18], ID and non-ID-based T-CH and D-CH protocols are proposed. In T-CH, though degree of rendezvous is increased, there is higher chance of collision in multiuser scenario as the sender and the receive use the same CH sequence. In D-CH, degree of rendezvous is very small with large value of MTTR, if both SUs have different IDs. In [9], although the authors have analyzed the feasible interference region for the PUs caused by the SUs in a CRN but there is no analysis about the percentage of spectrum usage and degree of rendezvous.

From the survey of all latest CH algorithms in CRN, it is observed that most of the proposed algorithms underperform in terms of degree of rendezvous, MTTR, maximum inter rendezvous interval (MIRI) and channel utilization. In this paper, efficient symmetric CH algorithms for the synchronous and asynchronous environment are designed and the main contributions of our work can be summarized as follows.

1) Our proposed protocols can work for any number of available channels.
2) The degree of rendezvous between any two SUs in our *SymSyn* CH Scheme is $\lessapprox \frac{N^2}{2}$, which is highest as compared to the existing protocols.
3) In our *SymSyn* protocol, we guarantee the successful rendezvous between two SUs with shortest interval of time such that MTTR $\leq \frac{N}{4}$.
4) A SymAsyn CH Scheme is introduced in which average degree of rendezvous is more than 50% with MTTR $\leq \frac{3N}{4}$.
5) The degree of rendezvous in our protocols increases with increase in the number of available channels, whereas this value normally decreases in case of other existing protocols.

III. Problem Formulation

Consider a CRN with both licensed PUs and unlicensed SUs in which fixed number of licensed orthogonal channels are accessed by all PUs. Depending on the absence of the PUs, SUs can access the licensed channels opportunistically. For simplicity, let N be the set of available channels for all SUs indexed from $\{0, ..., (|N|-1)\}$. Each SU is equipped with one half-duplex cognitive radio transceiver, which can have the capability to sense the spectrum holes. The entire network is divided into multiple slot indexes starting from $0, 1, ..., T_l - 1$, where T_l = length of the CH sequence. The CH pattern of a particular SU is determined by its CH sequence. The entire CH sequence of a SU comprises with two parameters, i.e., slot index, and

Fig. 1. Example of dynamic spectrum allocation in cognitive radio network (CRN).

channel number, which is defined as $\mathrm{CH\_seq} = (\mathrm{T_i}, \mathrm{C_j}), \forall \mathrm{i} \in [0, (\mathrm{T_l} - 1)]$ and $\forall \mathrm{j} \in [0, (|\mathrm{N}| - 1)]$, where the value of $T_l > |N|$. Therefore, any channel $c \in C_j$ will appear more than once in the hopping sequence. The CH sequence can be expressed as $\mathrm{CH\_seq} = \{(0, c_0), (1, c_1), \ldots, (i, c_i), \ldots, (T_l - 1, c_{T_l - 1})\}$, where $c_i \in [0, |N| - 1]$ represents the channel number visited by a SU at i-th slot index in a CH sequence. If two SUs hop to the same channel at the same slot index, they can listen to each other and use this channel as an operational channel for their data communication. Let CH1 and CH2 be the CH sequences of SU1 and SU2, respectively. The channel c can be the common channel for both the SUs, if $\exists\, i \in [0, T_l - 1]$, such that $(i, c) \in (\mathrm{CH1} \cap \mathrm{CH2})$. Let us consider a CRN that comprises both licensed PUs and unlicensed SUs with $|N|$ number of available channels. Considering an example as shown in Fig. 1, let number of available channels be five, i.e., $N = \{0, 1, 2, 3, 4\}$. The entire network is divided into multiple slot indexes indexed from $\{0, 1, 2, 3, 4\}$. From Fig. 1, it can be observed that some of the channels such as channel 1, 2, 3, and 4 are underutilized by the PUs at certain slots, thereby creating the spectrum holes. Hence, unlicensed SUs are embedded with smarter cognitive antenna to utilize the vacant portion of the channels more efficiently. Each SU follows a predefined CH sequence to hop from one channel to another and makes the rendezvous successful, as shown in Fig. 1.

A. Preliminaries

1) Degree of rendezvous: This metric is defined as the number of times two SUs meet with each other within a CH sequence.
2) Maximum time to rendezvous (MTTR): MTTR is the maximum time slots a SU takes to rendezvous within a hopping sequence. It is the worst case of TTR (time to rendezvous).
3) Average time to rendezvous (ATTR): Average time to rendezvous between two SUs is calculated as the average time required for SUs to rendezvous in one cycle. Hence, average of all possible TTRs is taken into account to calculate the ATTR.
4) Maximum inter rendezvous interval (MIRI): The MIRI is the maximum number of slot indexes between any two consecutive rendezvous between two SUs in a CRN. The MIRI is defined as the $\mathrm{MAX}\{(t_j - t_i) - 1\}$ (where t_i and t_j are the slot indexes of two consecutive rendezvous). The value of MIRI should be minimized to reduce the waiting period between the first and the next rendezvous.

B. System Model

In this section, we introduce a distributed mechanism to generate a general sequence set, as given in Algorithm 1, which is common to our proposed symmetric synchronous and asynchronous environment. In our protocols, CH sequence comprises with a *set of general sequences* denoted as $\mathrm{GS} = \{s_0, s_1, s_2, \ldots, s_{m-1}\}$, where $m < |N|$. SUs generate m number of general sequences by taking $|N|$ number of available channels into consideration. As given in Algorithm 1, first a *pivot element* is selected from the set of the available channels, which is then divided into three sets of channels as N_{front} *(N_f)*, N_{middle} *(N_m)*,and $\mathrm{N_{back}}$ (N_b) with help of that *pivot element*. A set of channel-shifting seeds (r) is obtained from the set N_f and N_b separately based on the rules given in the Algorithm 1. Then, the channel-shifting seeds are used for shifting the channels to generate the updated sets N_f and N_b. Finally, the general sequence set $\mathrm{GS} = \{s_0, s_1, \ldots, s_{m-1}\}$ is generated by calling the function $sequence\_geneartion$ as given in Algorithm 2 and is obtained by concatenating the updated sets N_f and N_b with N_m.

The generation of general sequences, as given in Algorithm 1, can be explained as follows. In line 2, a pivot element *p* is selected, whereas assignment of channels to set N_f, N_m, and N_b is made as given in lines 3 through 5. In lines 6 through 10, channel-shifting seed set *r* is calculated depending on the cardinality of $|N_f|$. In line 12, another sub function (*sequence_generation*) is called that returns the set of general sequences generated from the set N_f. The channel-shifting seeds for set N_b is calculated as given in lines 14 through 18. In line 20, again intermediate function (*sequence-generation*) is called to generate the general sequences from the set N_b. In line 21, both sets S_{N_f} and S_{N_b} are united to generate the final set of general sequence (GS). The set of general sequences GS that contains all generated general sequences is presented in line 22. In order to separate the generation of individual general sequences, a parameter $sequence\_bit$ is used. The channels of set N_f are executed if the $sequence\_bit == 0$ and the set N_b is executed if the $sequence\_bit == 1$.

As presented in Algorithm 2, the function $sequence\_generation$ represents all shifting and movements of the channels present in the set N_f and N_b. The elements of set N_f and N_b are assigned to the set N_a depending on the value of $sequence\_bit$ as given in lines 2 through 7. Lines 8 through 41 show how the general sequences are generated after applying each channel-shifting seed to both the set N_f and N_b. If channel-shifting seed is 0, all elements of set N_a are assigned to the intermediate set M_i as presented in lines 10 through 11. In line 13, calculation of $shifting\_channel$ is done. If channel-shifting seed is not 0, the codes given in lines 12 through 32 are executed. The lines 12 through 32 represent the anticlockwise movement of the channels after applying individual channel shifting seeds. The resulting channels generated due to anti-

Algorithm 1: General sequence generation.

Input: N : Set of available channels in the CRN.
Output: GS : Set of general sequences.
Notations:
N_f : Set that stores front segment of channels of set N;
N_b : Set that stores back segment of channels of set N;
N_m : Set that stores the pivot channel only;
$sequence\_bit$: Sequence bit that signifies the execution of set N_f or N_b; r : Set that stores the channel-shifting seeds; p : *pivot_element*; S_{N_f}, S_{N_b}: Sets that stores generated general sequences temporarily from set N_f and N_b; $N[]$: An array of channels.

1: Initialize $N_f = N_m = N_b = r = GS = S_{N_f} = S_{N_b} = \phi$;
2: $p = \lfloor |N|/2 \rfloor$;
3: $N_m = pth\ element\ of\ set\ N = \{N[p-1]\} =$ pivot channel;
4: $N_f = \{0, 1, \ \dots, \ N[p-2]\}$;
5: $N_b = \{N[p], \ \dots, \ |N| - 1\}$;
6: **if** $(|N_f|\%2 == 0)$ **then**
7: $\quad r = \{0, 2, 4, \ \dots, \ (|N_f| - 2)\}$;
8: **else**
9: $\quad r = \{0, 1, 3, \ \dots, \ (|N_f| - 2)\}$;
10: **end if**
11: $sequence\_bit = 0$; //Is taken for N_f
12: $S_{N_f} = sequence\_generation(N, N_f, N_m, N_b, r, sequence\_bit)$;
13: $r = \phi$;
14: **if** $(|N_b|\%2 == 0)$ **then**
15: $\quad r = \{0, 2, 4, \ \dots, \ (|N_b| - 2)\}$;
16: **else**
17: $\quad r = \{0, 1, 3, \ \dots, \ (|N_b| - 2)\}$;
18: **end if**
19: $sequence\_bit = 1$; //Is taken for N_b
20: $S_{N_b} = sequence\_generation(N, N_f, N_m, N_b, r, sequence\_bit)$;
21: $GS = S_{N_f} \cup S_{N_b}$;
22: Return GS

clockwise movement are stored in the intermediate set M_i. General sequences are generated by combining the channels present in the set N_f, N_b, N_m and M_i depending on the $sequence\_bit$ as given in lines 33 through 38. In line 39, each generated general sequence is concatenated with the temporarily stored sequence S'. In line 42, the generated general sequences stored in S' are returned to the main function $general\_sequence\_generation$.

Let us consider an example, where $|N| = 9$ is the number of available channels for the SUs in a CRN. Hence, set of available channels can be indexed as $N = \{0, 1, \ \dots, \ 8\}$, which can be initially arranged clock-wise in a ring as shown in Fig. 2(a). Then, the pivot element p is calculated based on our algorithm, which can be $p = \lfloor \frac{9}{2} \rfloor = 4$ here. Thus, set N_m contains only the pivot channel i.e. $N_m = p$-th element of set N = 4th element of

Algorithm 2: $sequence\_generation$ function call.

Input: $N, N_f, N_m, N_b, r, sequence\_bit$;
Output: S' : Set that stores the general sequences temporarily generated from sets N_f and N_b.
Notations:
N_a : Set that stores elements of N_f or N_b temporarily depending on the values of $sequence\_bit$;M_i : Set that stores the intermediate sequences after using channel-shifting seeds; $initial\_channel$: Stores the first element of set N_a; $shifting\_channel$: Stores the shifting channel after applying channel shifting seeds; $r[]$: An array of channel-shifting seeds used to access individual channel-shifting seeds.

1: Initialize $N_a = M_i = s = S' = \phi$;
2: **if** $sequence\_bit == 0$ **then**
3: $\quad$ Assign all the elements of set N_f to N_a i.e., $N_a = N_a \cup N_f$;
4: **end if**
5: **if** $sequence\_bit == 1$ **then**
6: $\quad$ Assign all the elements of set N_b to N_a, i.e., $N_a = N_a \cup N_b$;
7: **end if**
8: $initial\_channel = N_a[0]$; $len_{N_a} = |N_a|$; $len_{N_f} = |N_f|$; $len_{N_b} = |N_b|$;
9: **for** $i = 0$ **to** $(|r| - 1)$ **do**
10: $\quad$ **if** $r[i] == 0$ **then**
11: $\quad\quad$ Assign all elements of set N_a to M_i i.e., $M_i = M_i \cup N_a$;
12: $\quad$ **else**
13: $\quad\quad$ $shifting\_channel = [initial\_channel + len_{N_a}] - r[i]$;
14: $\quad\quad$ Append the $shifting\_channel$ to set M_i, i.e., $M_i = M_i \cup shifting\_channel$;
15: $\quad\quad$ $shifting\_channel = shifting\_channel - 1$;
16: $\quad\quad$ **while** $shifting\_channel >= initial\_channel$ **do**
17: $\quad\quad\quad$ Append the $shifting\_channel$ to set M_i i.e., $M_i = M_i \cup shifting\_channel$;
18: $\quad\quad\quad$ $len_{N_a} = len_{N_a} - 1$;
19: $\quad\quad\quad$ $shifting\_channel = shifting\_channel - 1$;
20: $\quad\quad$ **end while**
21: $\quad\quad$ **while** $len_{N_a} \neq 1$ **do**
22: $\quad\quad\quad$ **if** $sequence\_bit == 0$ **then**
23: $\quad\quad\quad\quad$ $len_{N_f} = len_{N_f} - 1$;
24: $\quad\quad\quad\quad$ Append the $len_{N_f}th$ element of set N_f to set M_i i.e. $M_i = M_i \cup N_f[len_{N_f}]$;
25: $\quad\quad\quad$ **end if**
26: $\quad\quad\quad$ **if** $sequence\_bit == 1$ **then**
27: $\quad\quad\quad\quad$ $len_{N_b} = len_{N_b} - 1$;
28: $\quad\quad\quad\quad$ Append the $len_{N_b}th$ element of set N_b to set M_i i.e. $M_i = M_i \cup N_b[len_{N_b}]$;
29: $\quad\quad\quad$ **end if**
30: $\quad\quad\quad$ $len_{N_a} = len_{N_a} - 1$;
31: $\quad\quad$ **end while**
32: $\quad$ **end if**

Continoud.

    if $sequence\_bit == 0$ **then**
      $s = \{M_i \cup N_m \cup N_b\}$;
    end if
    if $sequence\_bit == 1$ **then**
      $s = \{N_f \cup N_m \cup M_i\}$;
    end if
    Concatenate each generated general sequence with the general sequence set $S' = S' \parallel s$;
    $M_i = s = \phi$
  end for
  Return S'

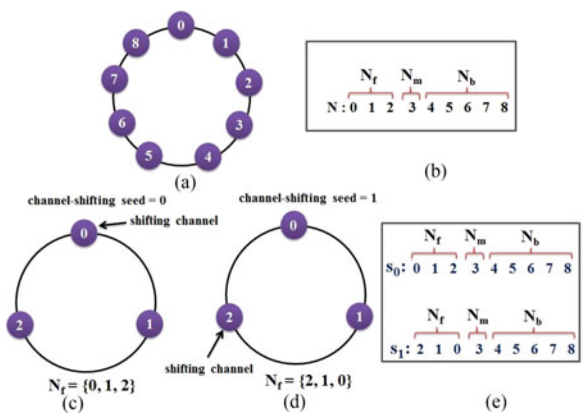


Fig. 2. Example of generating general sequences from set N_f.

set N $= \{3\}$. Then, the channels in the set N present before N_m are assigned to the set N_f, which can be $\{0, 1, 2\}$. Similarly, the channels in the set N present after N_m are assigned to another set N_b, which can be $\{4, 5, 6, 7, 8\}$, as shown in Fig. 2(b). All the channels in set N_f and N_b are also arranged in a ring. Each element of the set of channel-shifting seeds (r) is calculated from the sets N_f and N_b using Algorithm 1. For example, since, $|N_f|$ is 3 here, the set of channel-shifting seeds $r = \{0, 1\}$ according to our algorithm. Then, the corresponding *channel-shifting seeds* of N_f and N_b are applied to find the *shifting channels* in each set of N_f and N_b. Finally, a new set of *General Sequences* (GS) is generated taking the union operations between the channels present in the set N_f, N_m, and N_b.

In this example, when *channel-shifting seed* 0 is applied on set N_f, only clockwise movement is done without any shifting. Hence, after applying the channel-shifting seed $= 0$, an updated set N_f is generated without any change in the order, as shown in Fig. 2(c). When *channel-shifting seed* $= 1$ is applied on set N_f, *shifting channel* becomes 2, i.e., one anticlockwise shifting is done from the initial channel 0. After the shifting, an anticlockwise movement is carried out starting from the shifting channel 2. Thus, another set N_f is generated, which is given as $N_f = \{2, 1, 0\}$, as shown in Fig. 2(d). Finally, the *general sequences*s_0 and s_1 are generated by using the union operations between the channels present in the set N_m and N_b and taking the newly generated sets $N_f = \{0, 1, 2\}$ and $\{2, 1, 0\}$, respectively, as shown in Fig. 2(e). Similarly, as shown in Fig. 3, the set of channel-shifting seeds (r) is calculated from the set N_b, which contains channels $\{4, 5, 6, 7, 8\}$. Since, $|N_b| = 5$, $r = \{0, 1, 3\}$, as $(|N_b| - 2) = 3$. After applying each channel-shifting seed on the channels present in the set N_b, three new sets of N_b are generated as $N_b = \{4, 5, 6, 7, 8\}$ for channel-shifting seed $= 0$, as shown in Fig. 3(a), $N_b = \{8, 7, 6, 5, 4\}$ for channel-shifting seed $= 1$, as shown in Fig. 3(b), and $N_b = \{6, 5, 4, 8, 7\}$ for channel-shifting seed $= 3$, as shown in Fig. 3(c). Finally, general sequences are generated by performing the union operations between the channels of set $N_f = \{0, 1, 2\}$ and set N_m with each newly generated set N_b, as shown in Fig. 3(d). In summary, by using Algorithms 1 and 2 and as shown in Fig. 2, general sequences s_0 and s_1 are generated and as shown in Fig. 3, general sequences s_2, s_3, and s_4 are generated. Thus, for $|N| = 9$ as the number of available channels, 5 number of *general sequences* are generated and the *set of general sequences* (GS) can be GS $= \{\langle 0, 1, 2, 3, 4, 5, 6, 7, 8\rangle, \langle 2, 1, 0, 3, 4, 5, 6, 7, 8\rangle, \langle 0, 1, 2, 3, 4, 5, 6, 7, 8\rangle, \langle 0, 1, 2, 3, 8, 7, 6, 5, 4\rangle, \langle 0, 1, 2, 3, 6, 5, 4, 8, 7\rangle\}$.

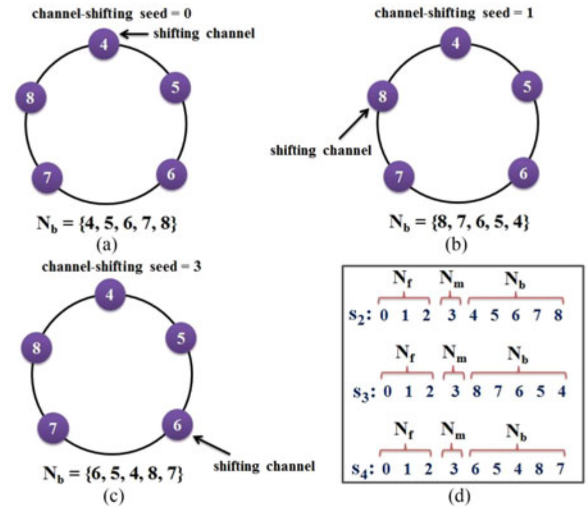


Fig. 3. Generation of general sequences from set N_b.

IV. Optimal Rendezvous CH Protocols

In this section, SymSyn and SymAsyn CH protocols are proposed, which can achieve optimal value in terms of degree of rendezvous, MTTR, ATTR, and MIRI. Detail algorithms of these protocols are described as follows.

A. SymSyn CH Protocol

Based on the standard definition of symmetric synchronous environment of CRN, each SU enters into the CRN at the same instant of time with availability of N licensed channels indexed from 0 to $|N| - 1$. The slot boundaries of all SUs are assumed to be aligned and the CH is started from the initial slot index 0. As described in the previous section, the general sequences $s_0, s_1, ..., s_{m-1}$ are generated, which are then shared by all SUs to generate the CH sequence. The algorithm for generating the CH sequence is given in Algorithm 3.

Let $|N| = 9$ be the number of available channels and GS $= \{s_0, s_1, s_2, s_3, s_4\}$ be the set of general sequences generated by two SUs, SU1 and SU2 by executing Algorithms 1 and 2. Then, each SU creates its own CH sequence by randomly choosing

Fig. 4. Rendezvous between two secondary users (SUs) in symmetric synchronous environment.

Algorithm 3: Generation of CH sequence for SymSyn Protocol.

Input: N: Set of available channels in the CRN.
Output: CH_seq: Channel Hopping sequence.
Notations
GS: Set of general sequences;
Hopping Sequence (HS): Set that stores permuted order of general sequences.
1: Initialize $HS = GS =$CH_seq$= \phi$;
2: $GS = general\_sequence\_generation(N)$;
3: $HS = HS \cup \{perm(GS)\}$;
4: $CH\_seq =< CH\_seq \parallel HS >$;

any one of the permutation of general sequences from the set GS. Let, s_0, s_1, s_2, s_3, and s_4 be the general sequences of SU1. As shown in Fig. 4, the general sequences of SU1 are s_0: $\langle 0, 1, 2, 3, 4, 5, 6, 7, 8\rangle$, s_1: $\langle 2, 1, 0, 3, 4, 5, 6, 7, 8\rangle$, s_2: $\langle 0, 1, 2, 3, 4, 5, 6, 7, 8\rangle$, s_3: $\langle 0, 1, 2, 3, 8, 7, 6, 5, 4\rangle$, s_4: $\langle 0, 1, 2, 3, 6, 5, 4, 8, 7\rangle$. Let, s_1, s_0, s_2, s_3, and s_4 be the general sequences of SU2. As shown in Fig. 4, the general sequences of SU2 are s_1: $\langle 2, 1, 0, 3, 4, 5, 6, 7, 8\rangle$, s_0: $\langle 0, 1, 2, 3, 4, 5, 6, 7, 8\rangle$, s_2: $\langle 0, 1, 2, 3, 4, 5, 6, 7, 8\rangle$, s_3: $\langle 0, 1, 2, 3, 8, 7, 6, 5, 4\rangle$, s_4: $\langle 0, 1, 2, 3, 6, 5, 4, 8, 7\rangle$. In this example, it can be observed that the degree of rendezvous between both SUs in a CH sequence is 41 out of 45 number of slot indexes. Then, the performance metrics of our proposed SymSyn CH protocol can be analyzed as follows.

1) Degree of Rendezvous: In our (SymSyn) CH Protocol, rendezvous occurs either in the channels of the set N_f (when channels of set N_b go for the shifting) or in the set N_b (when channels of set N_f go for the shifting) and there must be a guaranteed rendezvous among the channels present in the set N_m. Therefore, there must be rendezvous between two SUs.

Lemma 1: The maximum degree of rendezvous is $\leq [\{(m-2) \times |N|\} + \{2 \times (c_f + |N_m| + |N_b|)\}]$ for $|N| \geq 8$, else maximum degree of rendezvous is $\leq [\{(m-2) \times |N|\} + \{2 \times (|N_f| + |N_m| + c_b)\}]$, when both SUs permute different sets of general sequences. Value of c_b and c_f could be 1 or 2 based on the value of $|N_b|$ and $|N_f|$ is odd or even.

Proof: Let m number of general sequences be generated from $|N|$ number of available channels by SU1 and SU2. Let PS1 $= \{s_i, s_j, s_1, s_0, \ldots, s_{m-1}\}$ and PS2 $= \{s_j, s_i, s_1, s_0, \ldots, s_{m-1}\}$ be the permutation of general sequences chosen by SU1 and SU2, respectively. From these chosen general sequences, it can be observed that the order of general sequence from s_1 to s_{m-1} is same for both SUs, whereas the order of s_i and s_j is swapped between both permuted sets. It is known that the CH sequence of a SU $=$ the permuted general sequence set. Hence, the degree of rendezvous between the general sequences of PS1 and PS2 starting from s_1 to s_{m-1} is $(m-2) \times |N|$. Now, we need to analyze how many degree of rendezvous can occur within the first two general sequences of PS1 and PS2, i.e., $\{s_i, s_j\}$ and $\{s_j, s_i\}$. It can be calculated that the degree of rendezvous in both $\{s_i, s_j\}$ and $\{s_j, s_i\}$ are $(c_f + |N_m| + |N_b|)$ when $|N_f|$ is even and odd, respectively, where $c_f = 1,\ if\ |N_f|$ is odd and $2,\ if\ |N_f|$ is even. Considering all degrees of rendezvous, it can be calculated that maximum degree of rendezvous $= [\{(m-2) \times |N|\} + \{2 \times (c_f + |N_m| + |N_b|)\}]$. Similarly, for $|N| < 8$, the maximum degree of rendezvous depends upon the general sequences generated from set N_b. Hence, in this scenario, the maximum degree of rendezvous is $[\{(m-2) \times |N|\} + \{2 \times (|N_f| + |N_m| + c_b)\}]$ where $c_b = 1,\ if\ |N_b|$ is odd and $2,\ if\ |N_b|$ is even. For different set of permutations, degree of rendezvous could be $< [\{(m-2) \times |N|\} + \{2 \times (c_f + |N_m| + |N_b|)\}]$ or $[\{(m-2) \times |N|\} + \{2 \times (|N_f| + |N_m| + c_b)\}]$ depending on the number of channels. ■

Let us consider an example to explain degree of rendezvous. Let GS $= \{s_0, s_1, s_2, s_3, s_4\}$ be the set of general sequences generated when $|N| = 9$, where s_0 and s_1 are generated from the set N_f and s_2, s_3, and s_4 are generated from the set N_b. Let $\{s_0, s_1, s_4, s_2, s_3\}$ and $\{s_1, s_0, s_4, s_2, s_3\}$ be the permuted sets of general sequences chosen by SU1 and SU2, respectively. Here, SU1 and SU2 have the same order of permuted general sequences except s_0 and s_1, which are swapped. Hence, the maximum degree of rendezvous $= [\{(m-2) \times |N|\} + \{2 \times (c_f + |N_m| + |N_b|)\}] = [\{(5-2) \times 9\} + \{2 \times (1 + 1 + 5)\}] = 41$, where $|N_b| = 5$.

2) Average Time To Rendezvous (ATTR) and Maximum Time To Rendezvous (MTTR): The value of TTR varies with the ways channel-shifting seeds are used in the N_f or N_b part of a general sequence. If the initial general sequence of both SUs belongs to the general sequence generated by applying the channel-shifting seed on set N_b, then TTR $= 1$ as the channels present in the set N_f are same for all the general sequences generated from the set N_b. However, if the initial general sequences of both SUs are generated from the set N_f, the TTR varies with the of $|N_f|$ and the values of the channel-shifting seeds.

Lemma 2: If two SUs generate their first general sequence from the set N_f, one SU with channel-shifting seed $= 0$ and other SU with a nonzero channel-shifting seed, then ATTR $= \frac{\frac{|N|}{4}+2}{2}$ and MTTR $= \frac{|N|}{4}$.

Proof: Let us consider a set of available channels such that $|N| = 16$, where the channels present in the set N_f, N_b, and N_m are $\{0, 1, 2, 3, 4, 5, 6\}$, $\{8, 9, 10, 11, 12, 13, 14, 15\}$, and $\{7\}$, respectively. For the analysis of ATTR and MTTR, only the channels of set N_f are considered. Let FS be the set of all general sequences generated from the set N_f. Hence, FS $= \{s_0$: $\langle$0, **1**, **2**, **3**, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15$\rangle$, s_1: $\langle$6, 5, 4, **3**, 2, 1, 0, 7, 8, 9, 10, 11, 12, 13, 14, 15$\rangle$, s_2: $\langle$4, 3, **2**, 1, 0, 6, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15$\rangle$, s_3: $\langle$2, **1**, 0, 6, 5, 4, 3, 7, 8, 9, 10, 11, 12, 13, 14, 15$\rangle\}$. It is to be noted that general sequence s_0 is generated from the channel-shifting seed 0, while general sequences s_1, s_2, s_3 are generated from the nonzero channel-shifting seeds. Let SU1 considers the general sequence s_0 and SU2 considers either one of the general sequences s_1, s_2, or s_3. Hence, combinations of the general sequences could be $\{s_0, s_1\}$, $\{s_0, s_2\}$, and $\{s_0, s_3\}$ and from the above combinations, TTR could be $2, 3, ..., \frac{|N|}{4}$. Thus, the average TTR can be calculated as ATTR$=\frac{(2+3+\ldots+\frac{|N|}{4})}{\frac{|N|}{4}-1} = \frac{\frac{(\frac{|N|}{4}-1)\times(\frac{|N|}{4}+2)}{2}}{\frac{|N|}{4}-1} = \frac{\frac{|N|}{4}+2}{2}$. In this case, maximum TTR is the maximum value of all TTRs between SUs, i.e., MTTR$=$ MAX$(2, 3, ..., \frac{|N|}{4}) = \frac{|N|}{4}$. Particularly, if the combination of both SUs is $\{s_0, s_1\}$, then both SUs need to wait for four time slots, which is $\frac{|N|}{4}$ and it can also be true for any number of channels. ■

Lemma 3: If two SUs generate their first general sequence from the set N_f by using any nonzero channel-shifting seed, ATTR $= \frac{|N|}{2}$ and MTTR $= \frac{|N|}{2}$.

Proof: Let s_1: $\langle$6, 5, 4, 3, 2, 1, 0, 7, 8, 9, 10, 11, 12, 13, 14, 15$\rangle$, s_2: $\langle$4, 3, 2, 1, 0, 6, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15$\rangle$ be the initial general sequences of SU1 and SU2, respectively, which are generated from the set N_f by using a nonzero channel-shifting seed. It can be observed that the first rendezvous occurs at the pivot channel, which is common to all the sequences. Therefore, TTR $= \frac{|N|}{2}$. In this scenario, ATTR and MTTR are same as TTR. ■

Theorem 1: MTTR between any two SUs in *SymSyn* CH protocol is $= \frac{|N|}{2}$.

Proof: In Lemma 2, it is proved that MTTR $= \frac{|N|}{4}$, when the first general sequence of two SUs is generated by using a zero and a nonzero channel-shifting seed. In Lemma 3, it is proved that MTTR $= \frac{|N|}{2}$, when the first general sequence of the two SUs is generated by using a nonzero channel-shifting seed. Hence, from Lemma 2 and Lemma 3, MTTR of SymSyn CH protocol can be calculated as $MAX(\frac{|N|}{4}, \frac{|N|}{2}) = \frac{|N|}{2}$. ■

3) Maximum Inter Rendezvous Interval (MIRI): From the channels structure of general sequence, it can be visualized that there is no rendezvous between the general sequences in their N_f or N_b part, if they are generated by using a nonzero channel-shifting seed applied on the set N_f or N_b. If the order of the two consecutive general sequences for both SUs are generated from the channel-shifting seed other than 0, applied on set N_b and N_f, respectively. In this scenario, rendezvous is only possible in the N_m part of the second sequence. Hence, SUs need to wait for the entire length of the set N_b for the first sequence + entire length of the set N_f for the second sequence. Therefore, MIRI$= \lceil\frac{|N|}{2}\rceil + \lfloor\frac{|N|}{2}\rfloor \simeq |N|$. Considering an example for $|N| = 9$, where $\langle s_3, s_1, s_2, s_4, s_0\rangle$ and $\langle s_4, s_0, s_3, s_2, s_1\rangle$ are the CH sequences of SU1 and SU2, respectively, MIRI $= |N| - 2$.

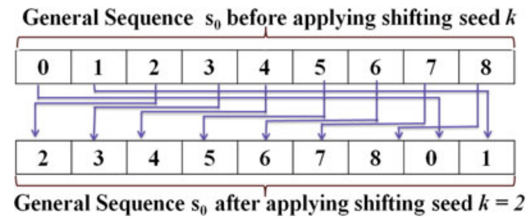


Fig. 5. Example of left circular Shift by $k = 2$.

B. SymAsyn CH Protocol

In asynchronous environment, SUs are assumed to enter into the network at different instant of time with knowledge about the number of available channels $|N|$ in the CRN. A time-slotted global clock system is considered, where duration of the global clock is equal to the length of the CH sequence. The local clock of each SU starts at the beginning of the global clock though it can enter into a channel at any instant of time represented in form of slot index. For the guaranteed rendezvous, the *left circular shift* (LCS) concept is introduced. The LCS operation for k bits on a sequence of elements is defined as the circular shift of all elements towards left with a shifting of k bits in clockwise. Consider a sequence s having length L_s such as $0, 1, ..., k, ..., L_s - 1$. After applying the LCS of k-bits on sequence s, the bits present in the sequence s become $k, k+1, ..., L_s - 1, 0, 1, k-1$. An example of LCS operation for $k = 2$ on the general sequence s_0 is shown in Fig. 5. After applying LCS operation for $k = 2$ bits, all channels of s_0 starting from channel 2 move by two positions towards left in clockwise, i.e., to channel 8. In SymAsyn CH protocol, each SU applies the LCS operation by k-bits in each general sequence and the value of k is decided based on its *entry slot index* (t_e). If $(\lfloor t_e\rfloor < |N|)$, $k = \lfloor t_e\rfloor$, else $k = (\lfloor t_e\rfloor \% |N|)$. The reason to introduce the concept of LCS is explained as below.

As shown in Fig. 6(a), there is no rendezvous between SU1 and SU2, if they enter into the network at different slot indexes. As shown in Fig. 6(b), LCS of 1-bit is applied on s_0 of SU1 and LCS of 2-bits is applied on the sequence s_4 of SU2. Here, the shifting parameter of 1-bit and 2-bits are nothing but the t_e of SU1 and SU2. From Fig. 6(b), we can see that there must be some rendezvous of channels between the general sequences s_0 and s_4. The same mechanism can also be applied to the rest of the general sequences for both the SUs.

It is to be noted that initially the set of GS of each SU is generated by applying Algorithm 1. Then, each SU carries out the LCS operation based on its k value, and updates the set of GS. Taking any permutation of those general sequences randomly,

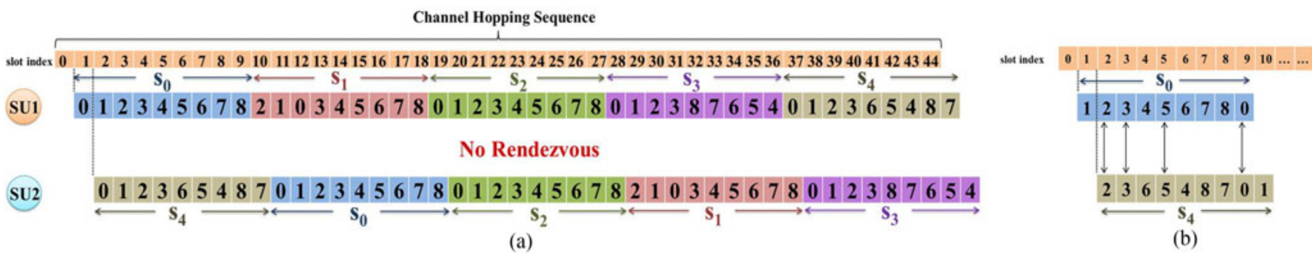


Fig. 6. Rendezvous analysis in symmetric asynchronous scenario.

Algorithm 4: Generation of CH sequence in Symmetric Asynchronous Environment.

Input: N: Set of available channels in the CRN.
t_e: Stores entering slot index of a SU.
Output: CH_seq: Channel Hopping sequence.
Notations:
GS: Set of general sequences; HS: Set that stores permuted order of general sequences; s: Set that stores individual general sequences before LCS; s_l: Set that stores individual general sequences after LCS; Seq: Set that stores the modified general sequences after LCS; k: Shifting seed; m: Stores the length of the GS.

Initialize $s = s_l = HS = GS = Seq = CH\_seq = \Phi$;
$GS = general\_sequence\_generation(N)$;
$m = |GS|$; $seq_{num} = 0$;
if $\lfloor t_e \rfloor < |N|$ **then**
  $k = \lfloor t_e \rfloor$;
else
  $k = \lfloor t_e \rfloor \ \% \ |N|$;
end if
while $seq\_num \leq m$ **do**
  Process individual *General Sequences* from set GS which will store in set s before LCS operation;
  for $i = 0$ **to** $(|N| - 1)$ **do**
    if $k \leq (|N| - 1)$ **then**
      $s_l[i] = s[k]$; $k = k + 1$;
    else
      $s_l[i] = s[n]$; $n = n + 1$;
    end if
  end for
  $\text{Seq} = \text{Seq} \| s_l$;
  $s = s_l = \Phi$;
  $seq\_num = seq\_num + 1$
end while
$HS = HS \cup \{\text{perm(Seq)}\}$;
$CH\_seq =< CH\_seq \parallel HS >$;

each SU generates its own CH sequence. The generation of CH sequence in SymAsyn environment is given in Algorithm 4.

Let $|N| = 9$ be the number of available channels and $\text{GS} = \{s_0, s_1, s_2, s_3, s_4\}$ be the set of general sequences generated by SU1 and SU2 by applying Algorithm 1. Hence, $\text{GS} = \{s_0 : \langle 0, 1, 2, 3, 4, 5, 6, 7, 8\rangle, s_1 : \langle 2, 1, 0, 3, 4, 5, 6, 7, 8\rangle, s_2 : \langle 0, 1, 2, 3, 4, 5, 6, 7, 8\rangle, s_3 : \langle 0, 1, 2, 3, 8, 7, 6, 5, 4\rangle, s_4 : \langle 0, 1, 2, 3, 6, 5, 4, 8, 7\rangle\}$. Let 3 and 20 be the t_e of SU1 and SU2, respectively. Hence, the shifting seed of SU1 and SU2 is $k = 3$ and $k = (20\%9) = 2$ (as $20 > |N|$), respectively. Now, each SU carries out the LCS operation on each general sequence based on its shifting seed k, and updates its GS. Now SU1 and SU2 create their own CH sequence by taking the random permutation of updated GS such as $< s_2, s_1, s_4, s_3, s_0 >$ and $\langle s_4, s_3, s_1, s_0, s_2\rangle$, respectively. The possible rendezvous between SU1 and SU2 are shown in Fig. 7.

1) Degree of Rendezvous: In SymAsyn CH protocol, degree of rendezvous varies with the permutations of different GS and the $clock\_offsets$ of the SUs. In our protocol, there is guaranteed rendezvous between any two general sequences irrespective of the misalignment of the CH sequences. The degree of rendezvous can be calculated based on the following lemma.

Lemma 4: In SymAsyn CH protocol, maximum degree of rendezvous between any two SUs is $\{(m \times |N|) - 2\}$, where m is the number of general sequences generated from the $|N|$ number of available channels.

Proof: Let us consider an example to explain Lemma 4. Let number of available channels be $|N| = 9$ and m number of general sequences are generated from this, which can be given as $\text{GS} = \{s_0{:}\langle 0, 1, 2, 3, 4, 5, 6, 7, 8\rangle, s_1{:}\langle 2, 1, 0, 3, 4, 5, 6, 7, 8\rangle, s_2{:}\langle 0, 1, 2, 3, 4, 5, 6, 7, 8\rangle, s_3{:}\langle 0, 1, 2, 3, 8, 7, 6, 5, 4\rangle, s_4{:}\langle 0, 1, 2, 3, 6, 5, 4, 8, 7\rangle\}$. By taking any pair of general sequences such as $\{s_0, s_1\}$ or $\{s_2, s_3\}$, there must be at least two nonoverlapping channels exist between them. If t_e of SU1 and SU2 is 1 and 4, respectively, SU1 and SU2 will go for the LCS with $k = 1$ and $k = 4$, respectively. Let, $\langle s_0, s_1, s_2, s_3, s_4\rangle$ and $\langle s_1, s_2, s_0, s_3, s_4\rangle$ be the CH sequences of SU1 and SU2, respectively, after taking the permutation. From the CH sequence of both SU1 and SU2, it is observed that rendezvous occurs in all slot indexes except the slot indexes where the pair of nonoverlapping channels are appeared such as $\{\langle 0\rangle, \langle 2\rangle\}$ and $\{\langle 2\rangle, \langle 0\rangle\}$. Hence, the degree of rendezvous in this case is $\{(5 \times 9) - 2\} = 43$, which can be generalized for any number of general sequences and available channels as $\{(m \times |N|) - 2\}$. ■

2) Average Time To Rendezvous (ATTR) and Maximum Time To Rendezvous (MTTR): In an asynchronous environment, the calculation of TTR purely depends on the value of t_e and the permuted set of GS. Therefore, we need to consider different cases considering t_e value and the channels present in the set N_f, N_m, and N_b. Let us consider two SUs generate their first general sequence from the set N_f, one SU with channel-shifting seed $= 0$ and other SU with a nonzero channel-shifting seed

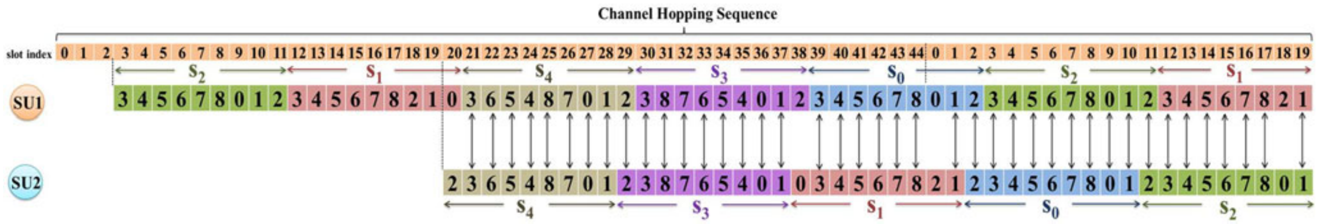


Fig. 7. Rendezvous between SUs where t_e of SU1 is 3 i.e., $k = 3$ and t_e of SU2 is 20 i.e., $k = 2$.

and let t_1, t_2 be the *entry slot index* (t_e) of both the SUs, respectively.

Case 1: If $(0 \leq (t_1 \text{ and } t_2) \leq \frac{|N|}{4})$ *and* $|N_f|$ *is odd:*

Let us consider an example when the GS are generated for $|N| = 16$ and $|N_f| =$ odd. Here, the set of general sequences GS $= \{s_0$:⟨0, **1**, **2**, **3**, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15⟩, s_1:⟨6, 5, 4, **3**, 2, 1, 0, 7, 8, 9, 10, 11, 12, 13, 14, 15⟩, s_2:⟨4, 3, **2**, 1, 0, 6, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15⟩, s_3:⟨2, **1**, 0, 6, 5, 4, 3, 7, 8, 9, 10, 11, 12, 13, 14, 15⟩, s_4:⟨0, 1, 2, 3, 4, 5, 6, 7, 8, **9**, **10**, **11**, 12, **13**, **14**, **15**⟩, s_5:⟨0, 1, 2, 3, 4, 5, 6, 7, 14, 13, 12, **11**, 10, 9, 8, **15**⟩, s_6:⟨0, 1, 2, 3, 4, 5, 6, 7, 12, 11, **10**, 9, 8, 15, **14**, 13⟩, s_7:⟨0, 1, 2, 3, 4, 5, 6, 7, 10, **9**, 8, 15, 14, **13**, 12, 11⟩$\}$. The channels present in the set N_f, N_b, and N_m are $\{0, 1, 2, 3, 4, 5, 6\}$, $\{8, 9, 10, 11, 12, 13, 14, 15\}$, and $\{7\}$, respectively. In the set GS, s_0 is generated from the set N_f by using the channel-shifting seed $= 0$, whereas s_1, s_2, and s_3 are generated from N_f by applying the nonzero channel-shifting seeds. If combination of sequences such as $\{s_0, s_1\}$ or $\{s_0, s_2\}$ or $\{s_0, s_3\}$ is taken, then there must be some overlapping of channels. Hence, $\text{TTR} = 1, 2, \ldots, \frac{|N|}{4}$. ATTR$= \frac{\frac{|N|}{4}+1}{2}$ and MTTR $=$ MAX (all TTRs between SUs) $= \frac{|N|}{4}$.

Case 2: If $(\frac{|N|}{4} < (t_1 \text{ and } t_2) < \frac{|N|}{2})$ *and* $|N_f|$ *is odd:*

Here, $\text{TTR} = 2, \ldots, \frac{|N|}{4}$, as there is no rendezvous in the N_f part of the sequence. In this case, rendezvous occurs at the channel present in the set N_m, i.e., at the channel number 7, as N_m is $\{7\}$. Hence, ATTR$= \frac{\frac{|N|}{4}+2}{2}$ and MTTR $=$ MAX (all TTRs between SUs)$=\frac{|N|}{4}$. The same scenarios can also be analyzed for $|N_f| =$ even as follows.

Case 3: If $(0 \leq (t_1 \text{ and } t_2) \leq \frac{|N|}{4})$ *and* $|N_f|$ *is even:*

Let us consider the GS generated from $|N| = 18$, where $|N_f|=$ even, i.e., $|N_f| = \{0, 1, 2, 3, 4, 5, 6, 7\}$. Here, the set of general sequences GS $= \{s_0$:⟨0, **1**, **2**, **3**, 4, **5**, **6**, **7**, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17⟩, s_1:⟨6, 5, 4, **3**, 2, 1, 0, **7**, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17⟩, s_2:⟨4, 3, **2**, 1, 0, 7, **6**, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17⟩, s_3:⟨2, **1**, 0, 7, 6, **5**, 4, 3, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17⟩, s_4:⟨0, 1, 2, 3, 4, 5, 6, 7, 8, 9, **10**, **11**, **12**, **13**, 14, 15, 16, 17⟩, s_5:⟨0, 1, 2, 3, 4, 5, 6, 7, 8, 17, 16, 15, 14, **13**, 12, 11, 10, 9⟩, s_6:⟨0, 1, 2, 3, 4, 5, 6, 7, 8, 15, 14, 13, **12**, 11, 10, 9, 17, 16⟩, s_7:⟨0, 1, 2, 3, 4, 5, 6, 7, 8, 13, 12, **11**, 10, 9, 17, 16, 15, 14⟩, s_8:⟨0, 1, 2, 3, 4, 5, 6, 7, 8, 11, **10**, 9, 17, 16, 15, 14, 13, 12⟩$\}$. The channels present in the set N_f, N_b, and N_m are $\{0, 1, 2, 3, 4, 5, 6, 7\}$, $\{9, 10, 11, 12, 13, 14, 15, 16, 17\}$, and $\{8\}$, respectively.

In this scenario, the values of TTRs are same with the case of $|N_f|$ is odd. Hence, $\text{TTR} = 1, 2, 3, \ldots, \frac{|N|}{4}$, ATTR $= \frac{\frac{|N|}{4}+1}{2}$ and MTTR$=\frac{|N|}{4}$.

Case 4: If $(\frac{|N|}{4} < (t_1 \text{ and } t_2) < \frac{|N|}{2})$ *and* $|N_f|$ *is even.*

If different combinations of general sequences are taken, then $\text{TTR} = 1, 2, \ldots, \frac{|N|}{4}$. Hence, ATTR $= \frac{\frac{|N|}{4}+1}{2}$ and MTTR$=\frac{|N|}{4}$.

However, if both the sequences of the set N_f are generated by using the nonzero channel-shifting seeds and the values of t_1 and t_2 are $0 \leq (t_1 \text{ and } t_2) < \frac{|N|}{2}$, there is no rendezvous in the N_f part of the general sequence, and rendezvous will occur among the channels present in the set N_m. Hence, $\text{TTR} = 2, 3, \ldots, \frac{|N|}{2}$. Therefore, ATTR $= \frac{\frac{|N|}{2}+2}{2}$ and MTTR $= \frac{|N|}{2}$. If initial general sequences of both SUs are generated from the set N_b and $0 \leq (t_1 \text{ and } t_2) < \frac{|N|}{2}$, then TTR $= 1$, as the channels present in the set N_f are same for all the general sequences generated from the set N_b. The same analysis of TTR, ATTR, and MTTR can be done for $\frac{|N|}{2} < (t_1 \text{ and } t_2) \leq (|N| - 1)$.

Theorem 2: MTTR in SymAsyn CH protocol is $\approx \frac{3\times|N|}{4}$.

Proof: Consider SU1 and SU2, which are associated with $clock\_offsets$ t_1 and t_2, respectively, where $t_2 > t_1$. The worst case of TTR arises, if $t_1 = 0$ and $t_2 = (\frac{|N|}{2} + 1)$ or due to the permuted order of general sequences. Let SU1 be at the general sequence s_i when SU2 enters to the network with initial general sequence s_j. If both sequences s_i and s_j are generated from the set N_b after applying the nonzero channel-shifting seeds, then no rendezvous occurs in the N_b part of both the sequences. Rendezvous occurs in the N_f part of next general sequence for SU1 with the N_f part of the sequence s_j for SU2. Hence, both SUs need to wait for the entire N_b part and maximum $\frac{|N|}{4}$ number of slot indexes in the N_f part. Therefore, MTTR can be calculated as $(\lceil\frac{|N|}{2}\rceil) + (\frac{|N|}{4}) \approx \frac{3\times|N|}{4}$. ■

3) Maximum Inter Rendezvous Interval (MIRI): In SymAsyn CH protocol, the MIRI between any two SUs is $\approx \frac{3\times|N|}{4}$, which is obtained from the analysis of MTTR.

V. Performance Evaluation

Performance of our protocols is evaluated by using OMNeT++ simulator IDE version 4.5 and is compared with some related works such as RCCH [1], ARCH [1], CACH [12], jump stay (JS) [13], L-QCH [14], FRCH [17], *EJS* [19] published in well-known journals. In the simulation, 50 PUs and 100 SUs are assumed to be deployed over an area of

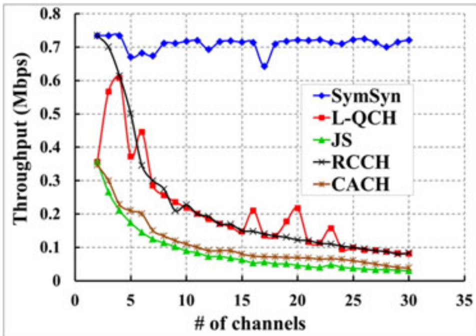


Fig. 8. Throughput versus number of channels.

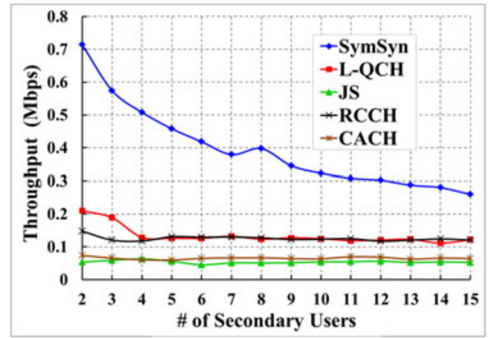


Fig. 9. Throughput versus number of secondary users.

1000 m × 1000 m. The entire network is divided into multiple number of time slots, where duration of each time slot is taken as 0.02 s. The study of the impact of number of channels on throughput, ATTR and % of rendezvous in the synchronous and asynchronous environment is carried out. The throughput is defined as the number of packets successfully transmitted between the sender and receiver per unit time.

A. Evaluation of Symmetric Synchronous Protocol

In Fig. 8, the impact of number of channels on the throughput is evaluated. In our SymSyn CH Protocol, there must be at least $\frac{N}{2}$ number of rendezvous between any two general sequences generated either from the set N_f or N_b. Besides, the degree of rendezvous between two SUs is $(m * |N|)$, i.e., 100% when they choose the same set of permuted general sequences, where m is the number of general sequences. Hence, the degree of rendezvous in SymSyn protocol increases with increase in the number of available channels, which gives more opportunity to the SUs for data communication. In RCCH, the throughput decreases gradually as the probability of rendezvous is very less, i.e., $\frac{2}{N}$. The throughput decreases exponentially, since only one channel acts as the rendezvous one, which may lead to the channel saturation problem. In L-QCH, the throughput decreases with the number of channels as only one rendezvous occurs per frame. As we know, JS protocol, contains one stay and one jump pattern. In jump pattern, rendezvous probability is very low and there is no rendezvous in the stay pattern, if both SUs can have different stay channels. Hence, the throughput of JS is very small as compared to other protocols.

The impact of number of SUs on the throughput is evaluated, as shown in Fig. 9. Throughput of L-QCH and CACH is very small due to the channel congestion problem as rendezvous occur only in a particular channel for the entire cycle or a frame. In RCCH, throughput decreases due to low probability of the rendezvous and very limited number of availability of alternate sequences, which increases the number of collision. In JS, the throughput is very less as the number of rendezvous is not uniform throughout the sequence. In SymSyn, the throughput in multiuser scenario is very high due to less chance of collision among SUs. In SymSyn, it is unlikely that two or more SUs can have same permutations of general sequences as plenty of permuted general sequences are available. Besides, each permutation can guarantee at least $\frac{N}{2}$ number of rendezvous between any two general sequences generated either from the set N_f or N_b. Due to constant number of channels considered in our simulation, the throughput decreases with increase in the number of SUs. However, our protocol outperforms in comparison to others.

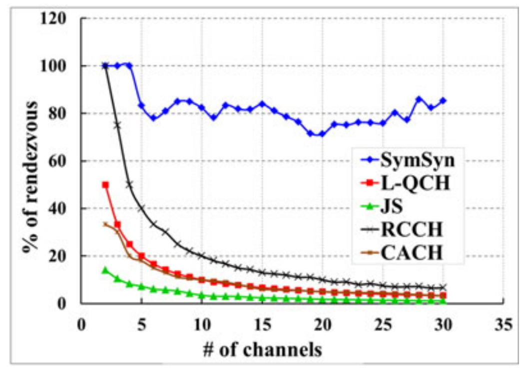


Fig. 10. % of rendezvous versus number of channels.

As shown in Fig. 10, it is observed that % of rendezvous in our protocol is very high as there must be at least $\frac{N}{2}$ number of rendezvous between any two general sequences generated from different channel-shifting seeds either from the set N_f or N_b. Besides, maximum degree of rendezvous in SymSyn protocol is $\{(m-2) * |N|\} + \{(2 \times (\lceil \frac{|N|}{2} \rceil + 1))\}$, which is much higher than other protocols, where m is the number of general sequences. Moreover, the degree of rendezvous between two SUs can be 100%, when they choose the same sets of permuted general sequences. In RCCH, percentage of rendezvous decreases exponentially with the increase in the number of channels as the probability of rendezvous is only $\frac{2}{N}$. In L-QCH, there is no increment in the number of rendezvous though length of the frame is increased with channel numbers. In JS, its very difficult to find a common available channel for the rendezvous due to its blind rendezvous nature. In CACH, the probability of rendezvous is very less, i.e., N out of $N \times (N+1)$ time slots.

The impact of number of channels on ATTR is shown in Fig. 11. ATTR of SymSyn in most of the scenarios is $\leq \frac{N}{8}$ with guaranteed rendezvous within $\frac{N}{2}$ numbers of slots. Hence, SUs need not to wait for more numbers of time slots

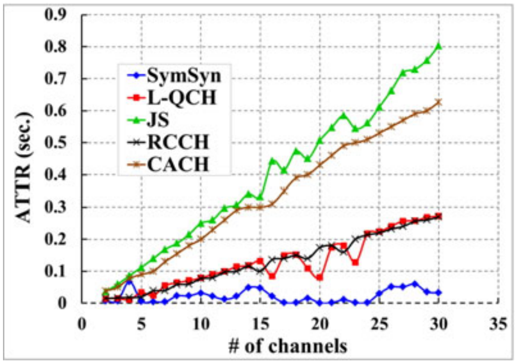


Fig. 11. Average time To rendezvous (ATTR) versus number of channels.

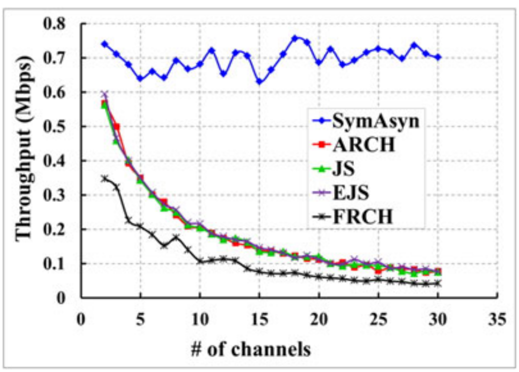


Fig. 12. Throughput versus number of channels.

for the rendezvous. In RCCH, the length of one cycle is N and number of rendezvous per cycle is 2. Hence, a SU has to wait for more number of time slots for its rendezvous with increase in the value of N, which increases the value of ATTR. In L-QCH, the frame length increases with increase in the channel number. The probability of rendezvous is very low per frame, which increases the value of ATTR. In CACH, value of ATTR increases as there is only one rendezvous within $N + 1$ intervals. In JS, it is very difficult to find one common channel within $3P$ time slots, which includes both jump and stay patterns. Therefore, the value of ATTR is very high as compared to other protocols.

B. Evaluation of SymAsyn Protocol

As shown in Fig. 12, it is clear that the throughput of ARCH, JS, and EJS is very low as compared to our SymAsyn due to less degree of rendezvous and large value of MTTR. In FRCH, the degree of rendezvous is very small with length of the CH sequence for which the number of transmissions between any two SUs is decreased. In our SymAsyn protocol, throughput is very high due to higher degree of rendezvous, i.e., the average degree of rendezvous is more than 50%, though it is not uniform due to permutation type and t_e of the SUs.

The impact of number of channels on % of rendezvous is presented in Fig. 13. In ARCH, the number of rendezvous is very small, i.e., only N out of N^2 time slots, which decreases the number of rendezvous with increase in channel numbers.

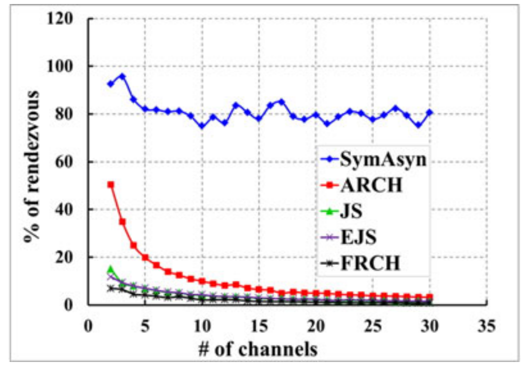


Fig. 13. % of rendezvous versus number of channels.

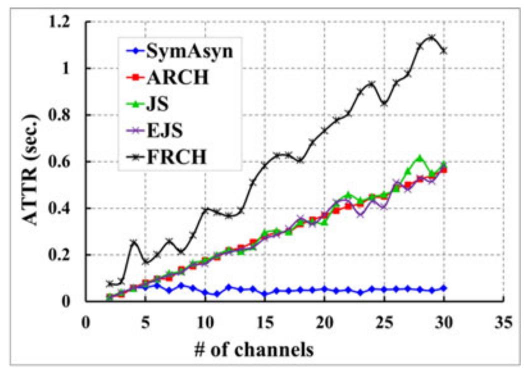


Fig. 14. Average time To rendezvous (ATTR) versus number of channels.

Degree of rendezvous in JS and EJS is not uniform as there is no guaranteed rendezvous within a bounded interval, which decreases the percentage of rendezvous. In FRCH, degree of rendezvous is very small as compared to the length of the CH sequence. Besides, control channel saturation problem arises, which degrades the performance. However, the degree of rendezvous in SymAsyn protocol increases with increase in the number of channels as well as is greater than the other protocols. The maximum degree of rendezvous of SymAsyn protocol is $\{(m \times |N|) - 2\}$, whereas the average degree of rendezvous is $> 50\%$.

The impact of number of channels on ATTR is evaluated, as shown in Fig. 14. In JS and EJS, the value of TTR is $\approx 3P$ and $\approx 4P$, respectively, by which the value of ATTR is increased. In ARCH and FRCH, value of ATTR is very large as the probability of the rendezvous is very low, i.e., $\frac{1}{N}$ and $\frac{N}{2\times(N+1)}$, respectively. Therefore, SUs have to wait more number of time slots. The value of ATTR in SymAsyn varies with the entry slot index of the SUs. However, the value is very small in most of the scenarios, i.e., $\leq \frac{N}{4}$.

C. Comparison Between SymSyn and SymAsyn Protocol

In order to compare the performance of our protocols SymSyn and SymAsyn, simulations are performed for the degree of rendezvous and ATTR, as shown in Figs. 15 and 16,

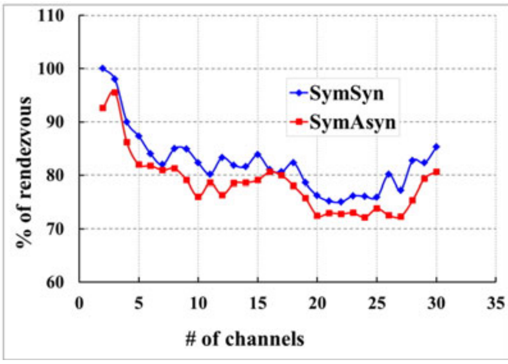


Fig. 15. Comparison of SymSyn and SymAsyn protocols in terms of % of rendezvous.

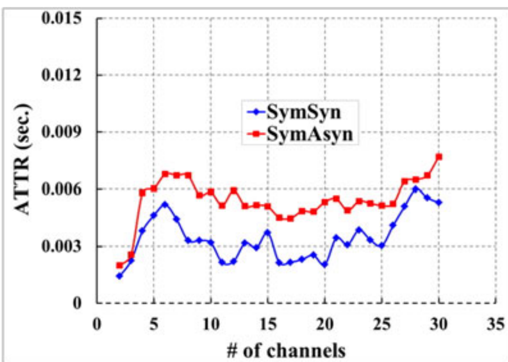


Fig. 16. Comparison of SymSyn and SymAsyn protocols in terms of ATTR.

respectively. The degree of rendezvous of both of our protocols is almost same and the values lies between 70% and 100%. As shown in Fig. 15, it can be observed that SymSyn shows better performance as the degree of rendezvous in SymSyn depends on the permutation of general sequences. In SymSyn, if two SUs choose the same set of general sequences simultaneously, the chance of degree of rendezvous is more. However, it varies with the entry slot index (t_e) of SUs in SymAsyn. As shown in Fig. 16, the value of ATTR for both of our protocols is very small as compared to other protocols. However, SymSyn performs little better in terms of ATTR, i.e., $\leq \frac{N}{2}$ than SymAsyn as ATTR varies with entry slot index of both SUs and lies within $0 \leq \frac{3\times N}{4}$ in SymAsyn.

VI. Conclusion

In this paper, two novel CH mechanisms are designed to optimize the degree of rendezvous, MTTR, ATTR, and MIRI in ICRN. Theoretical analysis of our proposed SymSyn and SymAsyn CH protocols are made to justify the correctness and to measure their performance. From the performance evaluation, it is observed that our protocols can achieve highest degree of rendezvous with minimum MTTR and MIRI. As compared to similar related works, our algorithms outperforms over other related protocols for both symmetric synchronous and asynchronous environment. Hence, our proposed protocols can be considered as the optimal solution for achieving higher degree of rendezvous with smaller value of MTTR and ATTR, which is highly essential for establishing the common control channel between two SUs in ICRN.

References

[1] G. Y. Chang, W. H. Teng, H. Y. Chen, and J. P. Sheu, "Novel channel-hopping schemes for cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 2, pp. 407–421, Feb. 2014.

[2] I. F. Akyildiz, W. y. Lee, M. C. Vuran, and S. Mohanty, "A survey on spectrum management in cognitive radio networks," *IEEE Commun. Mag.*, vol. 46, no. 4, pp. 40–48, Apr. 2008.

[3] T. M. Chiwewe, C. F. Mbuya, and G. P. Hancke, "Using cognitive radio for interference-resistant industrial wireless sensor networks: An overview," *IEEE Trans. Ind. Infromat.*, vol. 11, no. 6, pp. 1466–1481, Dec. 2015.

[4] P. K. Sahoo and D. Sahoo, "Sequence-based channel hopping algorithms for dynamic spectrum sharing in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 11, pp. 2814–2828, Nov. 2016.

[5] J. Miic and V. B. Miic, "Probabilistic vs. sequence-based rendezvous in channel-hopping cognitive networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2418–2427, Sep. 2014.

[6] L. Yu, H. Liu, Y. W. Leung, X. Chu, and Z. Lin, "Multiple radios for fast rendezvous in cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 9, pp. 1917–1931, Sep. 2015.

[7] F. Tang, L. Barolli, and J. Li, "A joint design for distributed stable routing and channel assignment over multihop and multiflow mobile ad hoc cognitive networks," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1606–1615, May 2014.

[8] Z. Gu, H. Pu, Q. S. Hua, and F. C. M. Lau, "Improved rendezvous algorithms for heterogeneous cognitive radio networks," in *Proc. 2015 IEEE Conf. Comput. Commun.*, Apr. 2015, pp. 154–162.

[9] M. Monemi, M. Rasti, and E. Hossain, "On characterization of feasible interference regions in cognitive radio networks," *IEEE Trans. Commun.*, vol. 64, no. 2, pp. 511–524, Feb. 2016.

[10] L. Chen, K. Bian, X. Du, and X. Li, "Multichannel broadcast via channel hopping in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 7, pp. 3004–3017, Jul. 2015.

[11] Y. Zhang, G. Yu, Q. Li, H. Wang, X. Zhu, and B. Wang, "Channel-hopping-based communication rendezvous in cognitive radio networks," *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, pp. 889–902, Jun. 2014.

[12] T. Y. Wu, W. Liao, and C. S. Chang, "CACH: Cycle-adjustable channel hopping for control channel establishment in cognitive radio networks," in *Proc. IEEE INFOCOM, IEEE Conf. Comput. Commun.*, Apr. 2014, pp. 2706–2714.

[13] H. Liu, Z. Lin, X. Chu, and Y. W. Leung, "Jump-stay rendezvous algorithm for cognitive radio networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 10, pp. 1867–1881, Oct. 2012.

[14] K. Bian, J. M. Park, and R. Chen, "Control channel establishment in cognitive radio networks using channel hopping," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 689–703, Apr. 2011.

[15] I. H. Chuang, H. Y. Wu, and Y. H. Kuo, "A fast blind rendezvous method by alternate hop-and-wait channel hopping in cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 10, pp. 2171–2184, Oct. 2014.

[16] Q. Liu, X. Wang, B. Han, X. Wang, and X. Zhou, "Access delay of cognitive radio networks based on asynchronous channel-hopping rendezvous and CSMA/CA MAC," *IEEE Trans. Veh. Technol.*, vol. 64, no. 3, pp. 1105–1119, Mar. 2015.

[17] G. Y. Chang and J. F. Huang, "A fast rendezvous channel-hopping algorithm for cognitive radio networks," *IEEE Commun. Lett.*, vol. 17, no. 7, pp. 1475–1478, Jul. 2013.

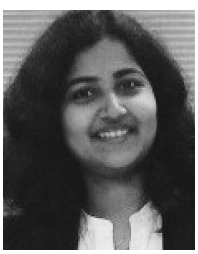
[18] G. Y. Chang, J. F. Huang, and Y. S. Wang, "Matrix-based channel hopping algorithms for cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 5, pp. 2755–2768, May 2015.

[19] Z. Lin, H. Liu, X. Chu, and Y. W. Leung, "Enhanced jump-stay rendezvous algorithm for cognitive radio networks," *IEEE Commun. Lett.*, vol. 17, no. 9, pp. 1742–1745, Sep. 2013.

Prasan Kumar Sahoo (SM'16) received the B.Sc. degree in physics (Hons.), the M.Sc. degree in mathematics both from Utkal University, Bhubaneswar, India, in 1987 and 1994, respectively, the M.Tech. degree in computer science from the Indian Institute of Technology (IIT), Kharagpur, India, in 2000, the first Ph.D. dgeree in mathematics from Utkal University, in 2002, and the second Ph.D. degree in computer science and information engineering from the National Central University, Taoyuan City, Taiwan, in 2009.

He is currently a Professor in the Department of Computer Science and Information Engineering, Chang Gung University, Taoyuan City, Taiwan. Since 2016, he has been an Associate Researcher in the Department of Cardiology, Chang Gung Memorial Hospital. He has worked as an Associate Professor in the Department of Information Management, Vanung University, Taoyuan City, from 2007 to 2011. He was the Director in the International Affairs Center, Chang Gung University, from 2013 to Jan. 2017. He was a Visiting Associate Professor in the Department of Computer Science, Universite Claude Bernard Lyon 1, Villeurbanne, France. His current research interests include big data analytic, cloud computing, and IoT.

Dr. Sahoo is an Editorial Board Member for the *International Journal of Vehicle Information and Communication Systems* and has worked as the Program Committee Member of several IEEE and ACM conferences. He was the Program Chair of ICCT, 2010.



Sulagna Mohapatra received the Master's degree in computer application from Chang Gung University, Taoyuan City, Taiwan, in 2015. She is currently working toward the Ph.D. degree in computer science and information engineering from the the Department of Electrical Engineering, Division of Computer Science and Information Engineering, Chang Gung University.

Her current research interests include wireless networks, particularly on channel hopping issues in cognitive radio networks.

Jang-Ping Sheu (F'09) received the B.S. degree in computer science from Tamkang University, New Taipei, Taiwan, in 1981, and the M.S. and Ph.D. degrees in computer science from National Tsing Hua University, Hsinchu, Taiwan, in 1983 and 1987, respectively.

He is currently a Chair Professor in the Department of Computer Science, National Tsing Hua University. He was the Chair in the Department of Computer Science and Information Engineering, National Central University, Taoyuan City, Taiwan, from 1997 to 1999. He was the Director in the Computer Center, National Central University, from 2003 to 2006. He was the Director in the Computer and Communication Research Center from 2009 to 2015, National Tsing Hua University. He was an Associate Dean in the College of Electrical and Computer Science from 2016 to 2017, National Tsing Hua University. His current research interests include wireless communications, mobile computing, and software-defined networks.

Dr. Sheu was an Associate Editor for the IEEE Transactions on Parallel and Distributed Systems and *International Journal of Sensor Networks*. He is currently an Advisory Board Member of the *International Journal of Ad Hoc and Ubiquitous Computing* and *International Journal of Vehicle Information and Communication Systems*. He received the Distinguished Research Awards of the National Science Council of the Republic of China in 1993–1994, 1995–1996, and 1997–1998. He received the Distinguished Engineering Professor Award of the Chinese Institute of Engineers in 2003. He received the K.-T. Li Research Breakthrough Award of the Institute of Information and Computing Machinery in 2007. He received the Y. Z. Hsu Scientific Chair Professor Award and Pan Wen Yuan Outstanding Research Award in 2009 and 2014, respectively. He received the Academic Award in Engineering from Ministry of Education and Medal of Honor in Information Sciences from the Institute of Information and Computing Machinery in 2016 and 2017, respectively. He is a member of Phi Tau Phi Society.